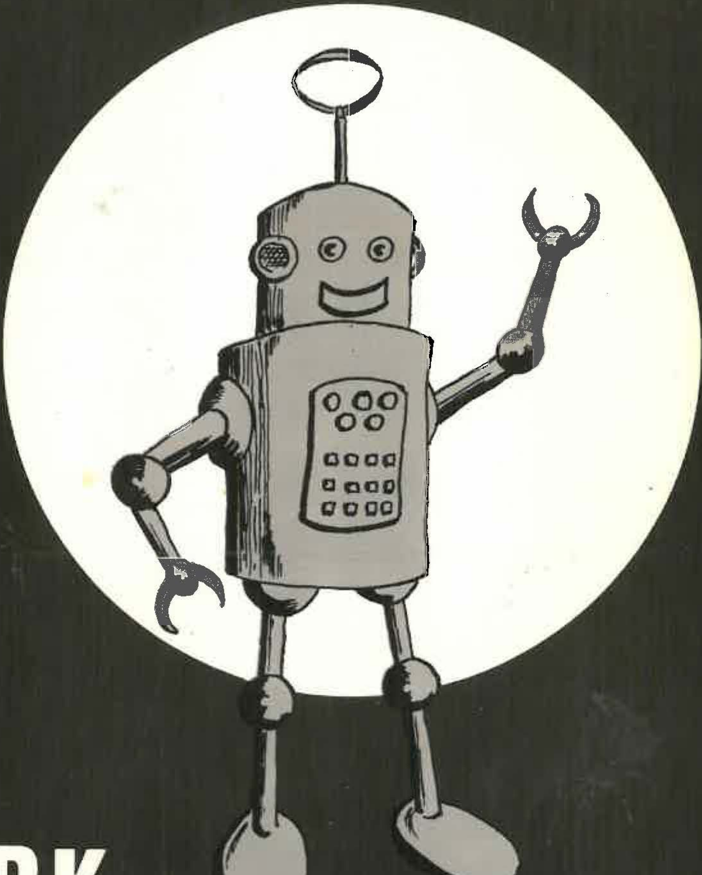
ROBOTICA PRACTICA PARA EL SPECTRUM



A.A.BERK

Robótica práctica para el Spectrum

Robótica práctica para el Spectrum

A. A. Berk

EDICIONES DIAZ DE SANTOS, S.A.
MADRID-BARCELONA

Título original: «Practical Robotics and Interfacing for the Spectrum»

ISBN: 0-246-12576-4 (edición en lengua inglesa).
ISBN: 84-86251-26-5 (edición en lengua española).
Depósito legal: M. 34.458-1985

EDITA: DIAZ DE SANTOS, S.A. 1.ª edición, 1985.
C/ Juan Bravo, 3-A. 28006 MADRID

PRODUCCION: ASEL, S.A.
C/ Clara del Rey, n.º 41. 28002 MADRID
Teléfonos: 416 62 47 - 65 98

Traductor: Alvaro Sánchez Medina
Revisado por: A.I.A.

Contenido

Prólogo

Probablemente la robótica es la materia más apasionante de todas las relacionadas con el mundo de la informática y de los ordenadores. Para la mayoría de las personas sin conocimientos técnicos especiales, un robot es algo así como una máquina de aspecto humano que realiza todas las funciones de una persona, aunque quizás de una manera no totalmente perfecta. Para el ingeniero, sin embargo, la palabra "robot" tiene un significado más amplio y a la vez más sencillo. Un robot, para el ingeniero, es una máquina que puede controlarse para realizar distintas actividades de una manera automática. Otros ingenieros no participan totalmente de esta definición porque añaden a lo anterior la capacidad de la máquina de enviar información de su actividad, realimentación, para que el control sea eficaz. Este diálogo, pues, entre el sistema de control y la máquina es lo que puede llamarse con bastante propiedad control inteligente, fundamental concepto para este tipo de ingenieros.

Otros ingenieros opinan que el robot debe poseer, imprescindiblemente, la capacidad de "aprender". De cualquier forma, la idea romántica del robot como androide, con el máximo de cualidades humanas, es en el fondo una aspiración secreta de todos, incluidos los ingenieros, que son personas prácticas. Por ello, aunque éstos opinen que esta figura humanoide está lejos aún de conseguirse, se deben investigar estas cualidades y la manera de copiarlas para ser ejecutadas por máquinas.

Este libro pretende proporcionar una visión general de las técnicas electrónicas que pueden ser realizadas, de una manera sencila, por cualquier persona que disponga de un Spectrum y tenga conocimientos de programación BASIC. El resultado no será un androide con un Spectrum como corazón, sino un conjunto de ideas, circuitos, y suficiente información para permitirnos explorar el área fascinante de la robótica.

El libro comienza con una introducción a la robótica, seguida por una explicación de la electrónica interna del Spectrum. Esto proporciona una información muy útil a cualquier poseedor de un Spectrum, aunque ya tenga una idea sobre el mismo. El siguiente capítulo facilita un ejemplo de aplicación sencillo utilizando el Spectrum. El capítulo 3 describe un interface especial para control de robots. Como ejemplo, se utiliza para el control del ratón robot, que se dirige hacia una fuente luminosa. El capítulo 4 explica los fundamentos de control de motores paso a paso, nos muestra la utilización de un sintetizador de voz y explica cómo monitorizar parámetros que varían de forma continua, tales como el nivel de intensidad luminosa. El capítulo 5 muestra algunos de los brazos robots más baratos del mercado y describe uno en particular asociado al Spectrum. Los apéndices describen los números binarios y la electrónica general e introducen al principiante las principales ideas las necesarias para que comprenda el libro.

Solamente se utiliza en el libro el lenguaje BASIC de programación, pero el que esté familiarizado con el código máquina será capaz de transferir estas ideas correctamente utilizando este lenguaje. Ningún proyecto de los descritos obligarán al lector a prescindir del uso normal del Spectrum para su realización. Los proyectos se describen minuciosamente para que sean perfectamente comprendidos. Con ello se pretende que los conocimientos adquiridos al realizarlos, sean la base que nos permita seguir investigando y diseñando nuestros propios proyectos.

Finalmente, hay varias personas que me han prestado ayuda y colaboración, proporcionándome material para este libro, y me gustaría mencionar a dos en particular: mi más sincero agradecimiento a mi esposa por su esfuerzo en animarme, y por la preparación de todos los documentos gráficos de manera que fueran inteligibles para la editorial. Gracias también a mi padre por su fundamental ayuda, sobre todo por la lectura y corrección, tanto de tipo general como técnica.

A.A. Berk

1
Introducción a la robótica y a la electrónica del Spectrum

¿Qué es la Robótica?

La ciencia ficción imaginada por el escritor Isaac Asimov, es probablemente la que ha ejercido más fuerte influencia en nuestra concepción de la *Robótica.* Sus historias nos hablan de un mundo en donde se construyen humanoides con todas facultades humanas inteligentes y muy pocos de sus defectos mecánicos. La complejidad del control de esas máquinas es confiada a un "cerebro de positrones", un bloque de material formado por millones de millones de sendas de datos, muy similares a los enlaces biológicos que forman el cerebro humano. Estos cerebros positrónicos serían mucho más eficientes que los nuestros en ciertas actividades, tales como la capacidad memorística y la velocidad de cálculo mental. Nunca se estropean y necesitan muy poca energía para su mantenimiento.

Además de este increíble mecanismo de control, los robots de Asimov poseen todas las características motoras humanas, pero con mucha más fuerza y eficiencia física general. También tienen la ventaja de que las partes averiadas se sustituyen fácilmente. Los robots de Asimov parecen humanos y se comunican en lenguaje normal.

La imagen de uno de estos robots ideales tal vez se nos aparezca así, aunque faltos de pensamiento. Sin embargo las influencias de esas ideas nos presentan una imagen del futuro donde el trabajo penoso de la vida diaria podría ser asignado a las máquinas, mientras nos deja libres para divertirnos con el fruto de su trabajo.

Desgraciadamente, antes de que este estado ideal pueda ser una realidad, es necesario que se produzcan muchos descubrimientos, tan revolucionarios como la invención del transistor. De igual modo, hoy por hoy, la robótica está influenciada por el deseo de recrear facultades humanas en las máquinas, sobre todo por la necesidad de comunicación de tales máquinas con los hombres para recibir instrucciones de ellos. Se arguye frecuentemente que hay que tratar de

hacer estas máquinas lo más parecidas posible al ser humano para minimizar barreras en la conexión "hombre-máquina".

El razonamiento anterior nos lleva a investigar en unas cuantas áreas específicas. En este libro se tratarán algunas de estas áreas de una manera general, tales como el control del movimiento de brazos artificiales, síntesis de voz, y locomoción en general. La visión es una de las cualidades más difíciles de tratar, aunque ya se ha trabajado en unidades de visión inteligente utilizando cámaras de televisión y detectores lumínicos, que se han desarrollado ya para ayudar a los exámenes físicos de obras de arte. Existen asimismo varias clases de sensores, como los táctiles, que utilizan microinterruptores, o los sensores de temperatura y presión, que utilizan dispositivos electrónicos.

Reproducir características humanas en máquinas es un proceso laborioso y detallado. Para ello, en primer lugar hay que estudiar estas características tal como son en el animal humano. A pesar de todo ya se están realizando dispositivos que en algún sentido son incluso más eficientes. Por ejemplo, actualmente se pueden producir palabras humanas cercanas a la perfección, cada palabra se almacena en una memoria, y puede ser recuperada por medio de una etiqueta electrónica específica, como veremos en el Capítulo 4. Automáticamente hemos producido un sistema que tiene capacidad recordatoria total, pero que no puede llevar a cabo ciertas particularidades o características humanas. Teóricamente, cada palabra debería poder emitirse con su propio timbre y tono para poder copiar la voz de diferentes tipos de individuos; bastaría poder variar unos cuantos parámetros electrónicos para poder producir una auténtica obra teatral a partir de este banco de datos de palabras.

Los sistemas electrónicos pueden ser diseñados para trabajar durante mucho tiempo sin fallos. Algunas averías se remedian simplemente reemplazando los componentes defectuosos. ¡Son tan flexibles los no humanos; se reparan tan fácilmente! En cambio, una simple gripe puede incapacitar a una persona durante días o semanas.

La ciencia de la robótica, basada en la mecánica y la electrónica, nos proporciona, en teoría, la posibilidad de dotar al producto final con todas aquellas características que nosotros desearíamos poseer. Desde luego, la más importante de estas características es la inteligencia. Este es un concepto tan complejo que sólo se ha estudiado superficialmente. Mucha gente opina que en este concepto deberían estar incluidas cualidades tales como la facultad de aprender, tomar decisiones, resolver problemas y cosas así. Todas estas ideas abstractas están siendo constantemente investigadas por los científicos dedi-

cados al diseño de ordenadores. Este libro trata los aspectos más puramente mecánicos de la robótica, y los próximos apartados describen algunas de las características humanas que, al menos, son parcialmente reproducibles.

La vista

El ojo es un instrumento notable todavía no plenamente comprendido. La manera en que podemos ver depende de la luz que llega de los objetos a nuestro alrededor, que, atravesando el globo ocular, es enfocada hacia la retina, en el fondo del ojo. La retina es algo parecido a una pantalla de proyección, compuesta de una estructura bioquímica compleja que convierte la imagen en impulsos nerviosos, los cuales son enviados al cerebro, donde son procesados. No obstante, el ojo no se limita simplemente a esta conversión, sino que también realiza algunos procesos con estos datos de imagen, independientemente del cerebro. Por ejemplo, detecta movimientos. Esta cualidad se centra fundamentalmente en los bordes del campo visual y ha dado pie a una expresión popular: "con el rabillo del ojo". Obsérvese que aunque no se pueda ver con nitidez en los límites del campo visual, los ojos se sentirán atraídos por algo que se mueva en esa zona.

Otro tipo de proceso que realiza el ojo es una cierta capacidad de reconocimiento de figuras. También este aspecto de la visión tiene reflejo en el lenguaje popular ("poner el ojo en..."). Esto significa literalmente que el sistema de la visión puede ser entrenado para reconocer aspectos específicos del campo visual de una manera inconsciente. Se pueden efectuar procesamientos sencillos de la imagen captada por una cámara de televisión. Los sensores de luz y oscuridad han sido ampliamente utilizados. En el Capítulo 3 se describe un proyecto que utiliza un sensor luminoso para controlar un ratón robot.

Llevamos muchos años detenidos en el conocimiento de la mayoría de los tipos fundamentales de visión inteligente. Las cosas más simples son dificilísimas de reproducir, por ejemplo, enseñar a una máquina a que mire una mesa y confiar en que la máquina reconozca una de cualquier tipo, en cualquier situación. Sin embargo, las cosas que podemos hacer son un importante comienzo y cualquiera puede experimentar con esas ideas, y quizás hacer que se de un paso adelante en el conocimiento de esta técnica. La mayoría de los aspectos de este campo de la técnica está en pañales, y el potencial por descubrir es enorme.

El oído

La detección y almacenamiento de sonidos, así como su análisis a diferentes frecuencias, está suficientemente estudiado. Podemos detectar sonidos de nivel bajísimo, al igual que otros, por su espectro de frecuencias, no podemos oirlos por estar fuera del rango de frecuencias del oído humano. La tecnología de fabricación de micrófonos es capaz de proporcionar una conversión de sonidos en ondas eléctricas tan fielmente, y reproducirlos electrónicamente con tal grado de perfección, que frecuentemente es imposible distinguir los sonidos reproducidos de los reales, si se dan las condiciones adecuadas. Por otro lado, el cerebro puede deducir la dirección desde la que llega un sonido, analizando la diferencia de tiempos con la que llega a uno y otro oído. Esto puede ser emulado por la electrónica, mejorando considerablemente la precisión de oído humano, y es parte de los estudios realizados para la técnica del radar o los dispositivos de escucha bajo el agua.

Estos aspectos, sin embargo, son los más elementales del sentido del oído. El oído humano es algo más que un simple detector de sonidos. Es capaz de distinguir modelos de sonidos, conjunto de sonidos, clasificándolos según la experiencia acumulada e identificándolos de manera inteligente. De nuevo observamos una forma de reconocimiento de modelos. La inteligencia del sistema acústico es otra forma fundamental de la comunicación. Por ejemplo, ¿cómo podríamos enseñar a una máquina a apreciar la música? Para contestar preguntas como ésta, es necesario mirar en nuestro propio interior. Los estudios sicológicos pueden sernos de gran ayuda. El arte de la reproducción de cualidades humanas sólo podrá conseguirse después de un profundo estudio de nuestra propia naturaleza.

El tacto

Existen varios niveles en el sentido del tacto. El primer nivel es simplemente la habilidad de sentir la presencia de un objeto. Esto se puede lograr electrónicamente por medio de unos cuantos sensores así como alcanzarse cualquier grado de precisión posicional.

El segundo nivel del sentido del tacto es el de ser capaz de distinguir entre diferentes tipos de objetos o tipos de superficies. También hay posibilidad de lograrlo electrónicamente de muchas maneras, con la ventaja de que las máquinas pueden ser construidas para trabajar en ambientes y lugares dañinos para las personas. En

este nivel puede incluirse también el "sentido" de la temperatura, de la presión, del movimiento, etc. La ciencia que trata de los sensores para fenómenos físicos está muy avanzada y podemos encontrar o fabricar un sensor para la mayoría de los usos.

El tercer nivel lo constituye la capacidad para detectar modelos, basada en experiencias previas. Esto es más que la simple medida del tamaño de un cuadro o el área de un círculo, cosas que podemos realizar ya con máquinas. Es algo parecido al reconocimiento de modelos (figuras) por medio de la vista, como cuando una persona puede reconocer una mesa, por ejemplo. Las personas no están particularmente capacitadas para reconocer objetos por el tacto, pero sí mucho más que cualquier máquina que podamos construir. Solucionando los problemas del reconocimiento de figuras en la visión, tendremos más facilidad de resolver los problemas del tercer nivel del tacto.

La voz

Hay dos aspectos fundamentales en el tratamiento de la voz humana, la *síntesis* y el *reconocimiento*. En el segundo de ellos nos encontramos de nuevo con el problema del reconocimiento de modelos. La síntesis de voz actualmente está bien estudiada, aunque ésto es sólo el nivel más superficial de la voz, Las personas utilizamos la voz para comunicarnos, de manera que una gran parte de la capacidad del cerebro está dedicada a esta función. Pasará algún tiempo todavía antes de que las máquinas puedan reconocer correctamente palabras pronunciadas por cualquier individuo y comunicarse con él de manera inteligente.

Actualmente, hay muchos tipos de sistemas de síntesis de voz en el mercado que, a su vez, se dividen en un número de clases. Dos tipos importantes son: *voz almacenada* y *síntesis por reglas*. El primero de ellos se basa en tomar una palabra completa, por medio de un micrófono, analizar las variaciones de voltaje producidas y almacenar estos datos de la forma más compacta posible. Esta palabra puede ser posteriormente reproducida enviando los datos almacenados a un circuito electrónico especial. Este tipo de sistema es barato y los fabricantes disponen de un gran almacén de palabras normales del cual se extraen para cualquier aplicación concreta.

La síntesis por reglas es un método para producir voz más inteligente y considerablemente más versátil. Se trata normalmente de un dispositivo que genera las palabras humanas a partir de las letras que

la componen, utilizando el código ASCII, con las palabras separadas por espacios. El sintetizador, entonces, trata de unir los sonidos de cada palabra utilizando un conjunto de normas y recompone el sonido de la palabra a partir de una serie de unos cincuenta sonidos básicos llamados *fonemas*, con los que se puede componer cualquier palabra. Teóricamente, utilizándo reglas similares a las que aprendíamos cuando éramos niños, seríamos capaces de formar la palabra. Así pues, tendríamos que introducir en la máquina reglas como "la p con la a, pa". El problema, desde luego, es que el lenguaje humano está lleno de casos especiales, que se han producido por nuestra manera muy humana de desarrollar gran profundidad y sofisticación sin una perfecta organización.

Con el fin de solventar este problema, la mayoría de los sintetizadores tienen un gran almacén de casos especiales. Así, cuando una palabra se le presenta para que la pronuncie, el sintetizador trata de adaptarla a una palabra normal en su memoria, para la cual tiene ya reglas y fonemas comprobados.

Este serviría bien como ejemplo de respuesta aprendida. Los sintetizadores pueden construirse fácilmente para utilizar la memoria de alguna forma cuando se presenta una nueva palabra. Un proceso típico puede ser el siguiente: se presenta la palabra, si la pronuncia mal, se conmuta el sintetizador al modo "aprendiz"; la palabra se presenta entonces en una fonética especial deletreándola a base de proporcionarle los fonemas, la máquina asimila el deletreo correcto, y almacena en su memoria los datos fonéticos junto con el deletreo. Desgraciadamente, hay algunos problemas. En primer lugar, el espacio de memoria y, en segundo lugar, el tiempo necesario para el proceso de recuperación o de búsqueda. Las memorias que pueden alterarse sin que se pierda su contenido cuando se les quita la alimentación, son voluminosas y caras. El segundo problema es aún más serio. Haría falta mucho tiempo para buscar una palabra en un gran fichero cada vez que se necesite una de ellas, dependiendo del tipo de memoria elegido. Si no se encuentra, la palabra debería ser sintetizada por reglas lo que supondría todavía más tiempo.

No obstante, todos estos problemas están ya a punto de ser solucionados. Seguro que la síntesis por reglas será el futuro de los sistemas de síntesis de voz. Actualmente estos dispositivos no proporcionan voces con la misma perfección con que lo hacen los sistemas de voz almacenada. Normalmente tienen un sonido "líquido", de máquina.

Otro nivel para la voz es el de los acentos e inflexiones, que dependen del contexto. La mayor parte del tiempo se emplearía en

examinar y entender la gramática para encontrar el procedimiento de expresión. Existen estupendos sintetizadores en el mercado, pero ninguno de ellos puede resolver realmente este problema a nuestra entera satisfacción, aunque alguno de ellos, de los más caros, nos hacen suponer que estamos muy cerca de resolver este problema.

El reconocimiento de la voz es, por supuesto, el problema más difícil de resolver en este campo. Se supone que la facultad de sintetizar la voz es paralela a la del reconocimiento de la misma. Pero el reconocimiento de la voz es otro ejemplo del reconocimiento de modelos, tan difícil de realizar en los campos de la vista y del tacto.

Hay algunos sistemas de reconocimiento sencillos que pueden entender unas pocas palabras, cuando la persona habla exactamente como el sistema ha aprendido, o bien cuando la máquina ha aprendido la forma de hablar del locutor. El proceso fundamental del mecanismo no está aún totalmente estudiado y parece que todavía tendremos que esperar algún tiempo antes de que el problema esté cerca de solucionarse.

El movimiento

Para mucha gente, en este campo es donde se deben concentrar principalmente los esfuerzos dirigidos a realizaciones robóticas. Entre los movimientos, se incluyen los de locomoción por medio de apéndices, que pueden ser desde ruedas a piernas mecánicas, así como brazos y dedos mecánicos que pueden realizar muchas funciones, como atrapar objetos, manejar herramientas, con posibilidad de graduar su fuerza.

Uno de nuestros atributos humanos es la capacidad de manipular objetos mecánicos con gran precisión, y es natural que seamos atraídos para realizarlos por medio de la electrónica. Hoy día se pueden producir movimientos de muchas maneras con el auxilio de la electrónica. Se pueden producir posicionamientos muy exactos y precisos de cualquier objeto, y éste es precisamente el corazón de cualquier sistema robótico.

El cuerpo humano realiza los movimientos contrayendo fibras musculares ancladas a palancas de tejido óseo. Las máquinas, en el momento presente, emulan esta actividad del cuerpo humano utilizando dispositivos basados en el electromagnetismo, en lugar de fibras y huesos. Tanta importancia tiene en robótica el electromagnetismo, o las máquinas que utilizan este principio, que en el siguiente apartado, antes de describir cualquier dispositivo electró-

nico, se explicará cómo puede utilizarse el electromagnetismo como principio para generar movimiento mecánicamente.

Dispositivos electromagnéticos

Existen máquinas eléctricas diferentes que producen movimiento, pero todas ellas se basan en el hecho de que si pasa una corriente eléctrica a lo largo de un conductor eléctrico, se produce un campo magnético predecible. Si en las cercanías del campo así producido se encuentra un imán, éste será atraído o repelido por este campo, mientras la corriente esté pasando. Cuando deja de pasar, cesa también la fuerza de atracción. La intensidad de la fuerza depende de varias cosas, tales como la intensidad de la corriente y la relación mecánica entre el conductor y el imán. Se puede incrementar esta fuerza incrementando la longitud del hilo por el que circula la corriente. Por el ejemplo, si el hilo está arrollado en un cilindro, el campo magnético a lo largo del eje del cilindro tiene mucha más intensidad que la producida por un trozo pequeño de hilo recto. Este tipo de bobina de hilo conductor se denomina *solenoide* (Fig. 1.1.). Si el interior del cilindro está compuesto de material magnético (hierro), la intensidad del campo será todavía mayor.

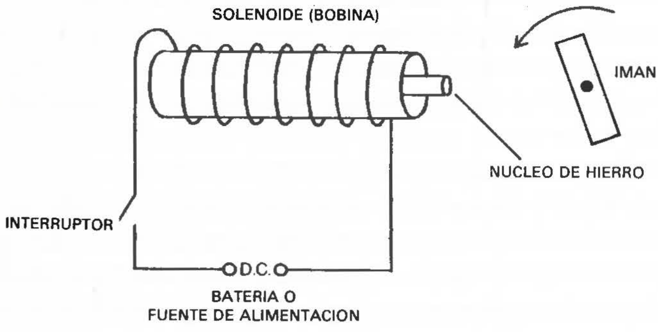


Fig. 1.1. Solenoide

Cuando se cierra el interruptor, circula una corriente continua (DC) por la bobina, y el imán se encontrará dentro del campo de fuerzas producido por el solenoide. La fuerza ejercida sobre el imán dependerá de la orientación del mismo, del sentido de paso de la corriente y de la dirección del arrollamiento del hilo en el cilindro. Si

el imán está sujeto en su centro de manera que pueda pivotar, girará hasta orientarse en una dirección determinada. De manera análoga, si el imán está fijo y es la bobina la que puede pivotar, aunque esto requiere el uso de contactos especiales, sería la bobina la que se orientaría con respecto al imán. Este tipo de rotación puede ser utilizado para mover engranajes y ejes y así crear movimiento para cualquier tipo de maquinaria. La bobina móvil es la base de los pequeños motores de corriente continua, alimentados con baterías, que se utilizan en juguetería y modelismo. Este tipo de motor podría emplearse para dirigir las ruedas directrices de un robot, como veremos en el Capítulo 3. Otra de las aplicaciones del solenoide es la producción de movimientos de vaivén para conseguir el relé magnético. El solenoide atrae una palanquita de hierro venciendo la fuerza de un pequeño muelle, cuando circula corriente por él. Cuando cesa la corriente, la palanquita vuelve a su sitio por la acción del muelle. La palanquita se utiliza para cerrar o abrir contactos eléctricos. En el Capítulo 2 podremos ver cómo el Spectrum puede gobernar un relé de este tipo.

Una de las principales necesidades en robótica es la de conseguir posicionamientos mecánicos de precisión. Esto se consigue por medio de un dispositivo llamado *motor paso a paso.* Estos motores se construyen de manera que su giro se divide en pasos de rotación perfectamente definidos, de tal manera que si se conectan a una palanca, o a una rueda a través de engranajes, el movimiento resultante es de una gran precisión. El motor paso a paso es el fundamento de la mayoría de los robots comerciales y son muy fáciles de controlar. En el Capítulo 4 veremos cómo el Spectrum puede controlar motores de este tipo utilizándolos, en este caso, para controlar los brazos de un robot.

El resto de este libro está dedicado a proporcionar tanta información como podamos para que se puedan realizar prácticas que ayuden a explorar más aspectos de la robótica, ciencia todavía en mantillas y de la que este libro sólo pretende introducir las ideas más fundamentales. En el próximo apartado se muestra el funcionamiento del Spectrum para ayudar a entender los proyectos que realizaremos posteriormente.

La electrónica del Spectrum

La clave indispensable para entender cómo utilizar su Spectrum para controlar dispositivos robóticos es comprender primero cómo

funciona el propio ordenador. El resto de este capítulo está dedicado a explicar el funcionamiento interno del Spectrum y de los microordenadores, en general.

La comprensión del Spectrum debe empezar con la lectura de los manuales que se entregan con la máquina. En el pequeño manual de introducción al Spectrum hay una fotografía del interior de la máquina a la que nos vamos a referir.

Dicha fotografía corresponde a la tarjeta "Versión Uno". La tarjeta "Versión Dos" tiene más cantidad de dispositivos electrónicos. De todas formas, dicha foto contiene suficiente información para nuestros propósitos.

La tarjeta mostrada se denomina *tarjeta de circuito impreso* (PCB). No hace mucho tiempo, los componentes electrónicos debían ser conectados entre sí utilizando hilo de cobre, hoy día los hilos de cobre se "imprimen" en la superficie de un substrato de un material, que suele ser fibra de vidrio, de manera que forman unas pistas de cobre planas sobre la superficie no conductora. Las pistas se recubren con una mezcla de estaño y plomo (soldadura) para protegerlas de la oxidación y para facilitar la soldadura de los componentes a ellas. Las pistas terminan normalmente en taladros interiormente metalizados, con objeto de servir de conexión de las pistas de un lado con las del otro. Los terminales metálicos de los componentes pasan a través de dichos taladros y se sueldan. Este tipo de PCB se denomina circuito impreso de taladros metalizados.

Como se puede ver, el ordenador tiene muchos componentes con sus patitas metálicas (terminales) insertadas en los agujeros del PCB. Una vez insertados los terminales en sus correspondientes taladros, deben soldarse. Esta operación, cuando se realizan grandes series no se efectúa normalmente con un soldador. Lo que se hace es transportar la placa con los componentes en un equipo de soldadura "por ola", el cual realiza todas las soldaduras, a un tiempo, después de un proceso de calentamiento de la placa, muy rápidamente.

Los principales componentes que pueden verse en la foto son los circuitos integrados (IC). Cada IC está compuesto por muchísimos componentes realizados por medio de métodos microfotográficos y fisicoquímicos sobre un único bloque o "chip" de silicio. El chip está sellado con un encapsulado de plástico. Los terminales están soldados al chip y salen al exterior de la cápsula de plástico y permiten conectar el chip con el resto del circuito. Por muy compleja que sea la misión del chip en el conjunto del circuito del que forma parte, sus terminales son el único contacto con el mundo exterior.

En el PCB también hay docenas de componentes pasivos, tales

como resistencias, para producir caídas de tensión, condensadores, utilizados para filtrar las tensiones de alimentación, principalmente, resistencias variables y condensadores variables para ajustes tales como los de la imagen de vídeo.

Támbién hay otros componentes importantes. Los diodos, que se utilizan en los circuitos lógicos o en la fuente de alimentación y los cristales de cuarzo, que se utilizan para estabilizar los osciladores que generan relojes de precisión para el control de tiempos, por ejemplo. En la fuente de alimentación se utiliza una pequeña bobina, en conjunción con un IC para regular el voltaje que alimenta los demás circuitos. Otros componentes que podemos encontrar son, por ejemplo, el altavoz, el modulador de VHF o UHF para proporcionar señales compatibles con el receptor de televisión, y varios tipos de conectores para conectar dispositivos externos a la máquina.

Veamos ahora esto con más detalle, pero todavía como una visión panorámica del ordenador. La figura 1.2. muestra el diagrama de bloques de la electrónica del Spectrum. Hay muchas interconexiones que se han omitido, para mayor claridad, aunque algunas se describirán con más detalle más adelante. Los datos electrónicos son simplemente unos y ceros y son materializados como señales de +5V (1 lógico) o 0V (0 lógico) en los conductores eléctricos. Ver los apéndices 1 y 2 para mayor información.

Existen también otros componentes muy importantes, como veremos a continuación.

Líneas de BUS

La Figura 1.2 muestra un número de dispositivos electrónicos conectados entre sí por un grupo de líneas. Todos los datos electrónicos y de control dentro de un ordenador no son más que un modelo o un conjunto de estados binarios de los conductores eléctricos. Muchos de esos conductores realizan una función común y pueden ser convenientemente agrupados formando *Buses*. Una parte del dato, que puede ser, por ejemplo, un número binario de 8 bits de un cálculo, o un carácter del código ASCII de algún texto, aparecerá inevitablemente en algún lugar del ordenador como un grupo de señales electrónicas en un instante del tiempo. En general, todos esos bits aparecerán al mismo tiempo, en *paralelo*, en un grupo de líneas de datos. Si fuéramos suficientemente rápidos, durante el cálculo, podríamos congelar el estado de este bus, conectar un voltímetro en

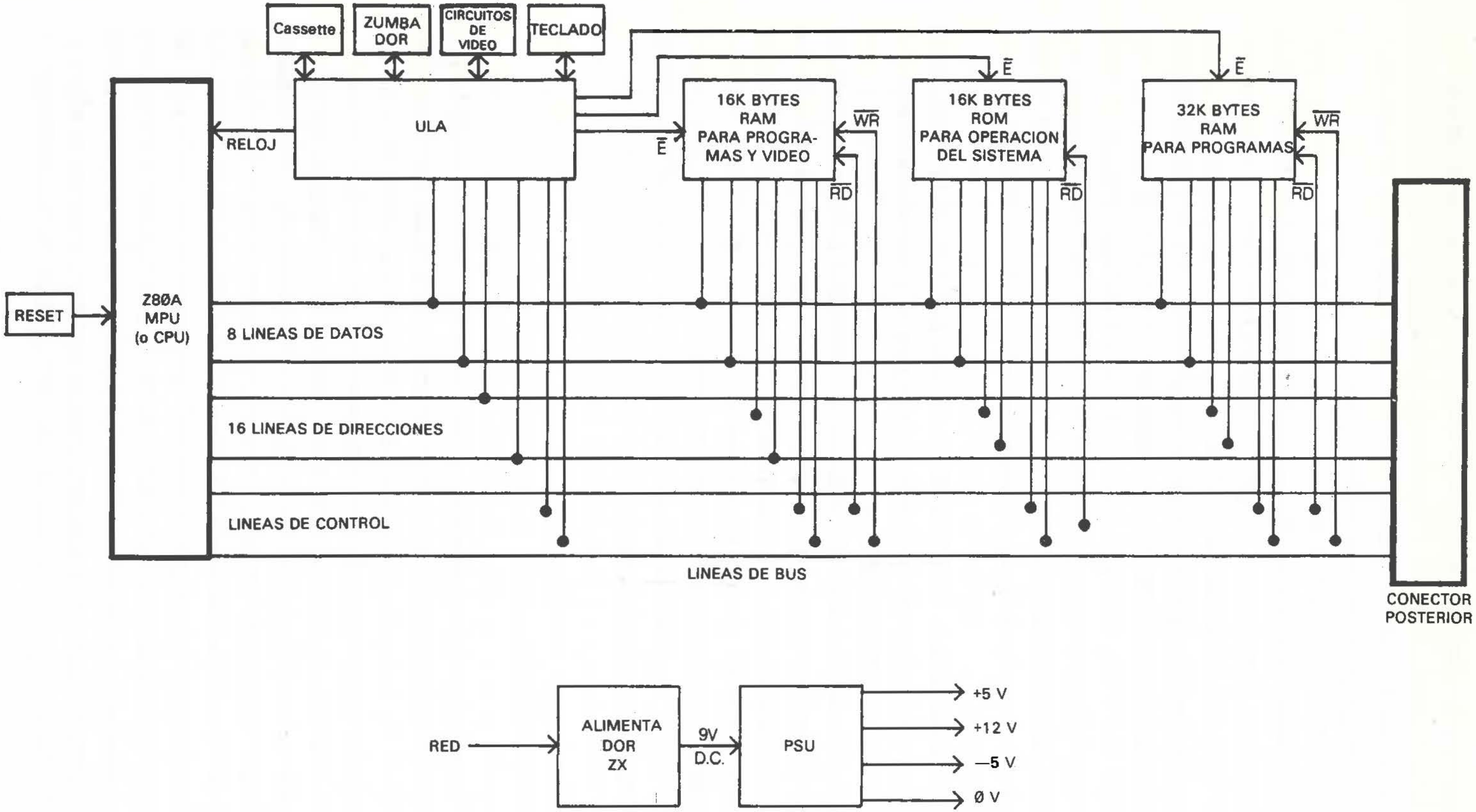


Fig. 1.2. Diagrama de bloques del Spectrum

cada línea de datos y leer el número binario como un conjunto de unos (+5V) y ceros. Esto es lo que efectivamente hacen ciertos aparatos diseñados especialmente para ello y que se utilizan para la reparación y diseño de nuevos sistemas. Se denominan *analizadores lógicos*, pues pueden averiguar los "estados lógicos" (unos y ceros) en un grupo de líneas en un instante determinado, congelando la actividad de las mismas para visualizarla y analizarla.

Puesto que los bits del número binario mencionado aparecen simultáneamente en las líneas del bus, diremos que éste es un *bus paralelo* y que se ha producido una comunicación paralelo. En los casos en los cuales los bits del número se envían uno por uno a través de una sola línea, diremos que se produce una comunicación *serie*.

Hay tres buses en un microordenador típico, el *bus de datos*, el *bus de direcciones* y un grupo de líneas de carácter más indefinido, llamado *bus de control*. El bus de datos tiene 8 bits de extensión y el bus de direcciones tiene 16. Todos los dispositivos del ordenador están conectados a esos dos buses y es importante, por lo tanto, que no se activen dos de ellos al mismo tiempo, o colisionarían. Cada dispositivo tiene una etiqueta electrónica o *dirección*. Esta dirección es llevada a lo largo del bus de direcciones para alertar y activar un solo dispositivo cuando sea necesario. Esta función la realiza el CPU (unidad central de proceso), o microprocesador, generando esta información de dirección y asegurando así que se está usando el dispositivo correcto en cada momento. Esto quedará mejor explicado en la descripción que haremos de la ULA.

Microprocesador Z80A

Este es el microprocesador propiamente dicho y el verdadero cerebro del ordenador. Se le denomina a veces MPU (unidad de microprocesador), aunque tal vez, como parte integrante de un ordenador sea más común llamarle CPU (unidad central de procesos). Su principal atributo es sencillamente que puede recabar instrucciones de la memoria en forma de datos binarios, por medio del bus de datos, y ejecutarlas de manera secuencial. Este conjunto de instrucciones se denomina *programa* y esta cualidad de recabar datos de la memoria es una de las actividades fundamentales que ocurren en la máquina. Es un ejemplo de transferencia de datos en paralelo, que debe realizarse controlando el conjunto de líneas de bus para obtener los estados lógicos necesarios en el momento oportuno.

El CPU elegido para un sistema determina tanto la electrónica como la manera de trabajar la máquina. El Z80A es un microprocesador de 8 bits típico. Todo lo que aprendamos acerca de este dispositivo nos servirá en gran manera para la comprensión de otros microprocesadores. Se dice que el Z80A es un microprocesador de 8 bits porque su bus de datos es de 8 bits y porque puede procesar los 8 bits al mismo tiempo.

Para recabar datos e instrucciones de la memoria, el CPU envía hacia fuera direcciones por el bus de direcciones, al tiempo que controla los dispositivos de memoria electrónica por medio de un conjunto de líneas de control. Como resultado espera recibir datos binarios por sus líneas de datos. Las líneas de datos son bidireccionales, lo que permite además al CPU almacenar datos en la misma forma.

El CPU determina en qué momento las líneas de datos tienen carácter de entrada (IN) o de salida (OUT). Utiliza para ello dos líneas de control, que informan a la memoria de este hecho. Estas líneas se denominan línea RD y WR, como se muestra en la figura 1.2. Estas líneas las lleva el CPU al nivel 1 o al cero, cuando conviene. Cuando la línea RD está en el estado 0, la memoria sabe que ésto quiere decir que va a ser *leída* por el microprocesador, y éste ajusta sus líneas de datos para que sean líneas "in", por las cuales le llegarán los datos. Cuando la CPU hace que la línea WR tome el valor 0, la memoria sabe que la CPU quiere *"escribir"* en ella. La CPU asegurará cuidadosamente que sólo una de las dos líneas esté a nivel bajo en todo instante para prevenir colisiones. Nótese que los símbolos de estas líneas se representan con las letras anteriores y unas "barras" sobre ellas. Este es un convenio adoptado para las líneas que funcionan de este modo: la función de la línea se activa si está a nivel lógico cero y viceversa. De la misma manera, si los símbolos de una línea no llevan estas barras, su función se activará cuando esté a nivel lógico 1.

Esta explicación sobre las líneas RD y WR la hemos proporcionado como ejemplo de la cantidad de controles que continuamente mantiene la CPU sobre el sistema. También es un buen ejemplo de un típico control de líneas.

Otro aspecto importante de la CPU es el conjunto de instrucciones que puede ejecutar. Estas instrucciones están almacenadas en localizaciones de memoria como modelos electrónicos (conjuntos ordenados de ceros y unos). La CPU debe tener constancia exacta en cada momento de lo que está haciendo, para no mezclar estas instrucciones con los datos propiamente dichos, que se almacenan en

el mismo tipo de localización de memoria usando el mismo tipo de modelos electrónicos. La interpretación exacta de cada instrucción la decide el diseñador de la CPU. El conjunto de «códigos de operación», como son llamados, está proporcionado como números en binario o hexadecimal (ver Apéndice 1). Cada uno de estos números puede ser almacenado electrónicamente, y cuando los solicita el CPU de la memoria, esto hace que la CPU realice un conjunto definido de actuaciones. El lenguaje de estos códigos de operación se llama *código máquina,* y es perfectamente posible aprender cómo funciona y programar el ordenador utilizando este lenguaje. La programación en código máquina es un método muy potente de programar un ordenador, y los programas en código máquina se ejecutan considerablemente más deprisa que los programas en Basic. Sin embargo, la programación en código máquina es también muy tediosa. El Spectrum hace más fácil la programación por medio de un programa especial llamado *intérprete.* Esto lo controla el ordenador y permite al usuario introducir el programa por medio de instrucciones en Basic. El intérprete, entonces, convierte éstas en código máquina, para que el Z80 pueda entenderlas.

RAM

RAM (random access memory) significa *memoria de acceso aleatorio,* y es un tipo de memoria cuyo contenido puede cambiarse a voluntad para almacenar datos y programas, perdiéndose cuando se apaga el ordenador. Este tipo de memoria se denomina *volátil.*

Nótese que existen dos bloques de RAM en el Spectrum, el bloque de 16K (bytes), que es el que encontramos en la máquina de 16K, y el bloque de 32K (bytes) que, añadiéndoselo, la convierte en una máquina de 48K. En la Versión Uno la memoria extra se añade por medio de un circuito impreso separado, mientras que en la Versión Dos los chips pueden enchufarse directamente en los lugares que existen al efecto en el propio circuito impreso original.

El bloque de 16K contiene la memoria de vídeo o 'VideoRAM', cuya misión es la de mostrar los datos almacenados en la pantalla del televisor. Además, el bloque de 16K de RAM contiene las variables del sistema y otros conjuntos de datos que necesita el ordenador para poder funcionar. Así pues, sólo una fracción de esta RAM está disponible para albergar programas de Basic y datos. El bloque de 32K, en cambio, puede utilizarse completamente para almacenamiento de programas y datos.

Cada bloque de RAM posee una línea de conexión que se reconoce en los diagramas esquemáticos por medio de la letra E con una barra superior, conectada a la ULA. Cuando esta línea se encuentra en estado lógico 'bajo', está activado el bloque RAM. Es la ULA la que se encarga de colocar este terminal a nivel bajo cuando reconoce una dirección en el bus de direcciones que corresponde a una zona dentro de ese bloque de memoria. Al mismo tiempo los otros bloques se desactivan, siendo, en efecto, 'desenchufados' electrónicamente del sistema. Este estado 'desenchufado' es denominado 'triestado' o de 'alta impedancia', y la facultad de los circuitos de habilitar este tercer estado es uno de los avances más importantes en la moderna tecnología de ordenadores. Esta es la razón de que todos los bloques puedan soldarse eléctricamente a las mismas líneas del bus, sin interferencias.

ROM

La ROM (read only memory), o *memoria de sólo lectura*, es una memoria similar a la RAM, excepto que su contenido no puede ser alterado y no desaparece aunque se quite la tensión de alimentación. El circuito integrado ROM es un bloque completo de memoria. También posee una línea E-barra conectada a la ULA puesta a nivel activo (bajo) siempre que la CPU demanda una dirección que está contenida en algún lugar dentro de la matriz de direcciones ROM. El chip, entonces, decodifica su propia dirección interna para asegurar que el dato leído de ella se ha tomado de la posición correcta dentro de la matriz.

La información contenida dentro de la ROM se denomina *firmware* (microprograma), y consiste en las instrucciones que permiten al usuario introducir programas en Basic y ejecutarlos. Aquí se encuentra el intérprete que hemos mencionado anteriormente. La ROM también contiene todas las rutinas que permiten utilizar el teclado y leer de manera comprensible el texto de la pantalla. Este tipo de programación contenida en la ROM es la que permite al usuario programar el ordenador sin necesidad de que éste entienda su electrónica interna. La enorme complejidad del control electrónico, efectuado millones de veces cada segundo, se le oculta al usuario.

La ROM está especialmente diseñada por Sinclair para el Spectrum. No sería posible enchufar el chip a otra máquina y usarla como se hace en el Spectrum. Cada máquina distinta tiene distinta

configuración electrónica y necesitan programas diferentes para su control. La más importante de estas diferencias reside en el mapa de memoria, que analizaremos en breve.

ULA

La ULA (uncomited logic array) es un dispositivo que comprende en un solo chip un conjunto heterogéneo y específico de circuitos electrónicos que realizan varias funciones y que ha sido diseñado especialmente, por encargo. Cuando una compañía diseña un circuito electrónico de suficiente complejidad, requiriendo, tal vez, docenas de circuitos integrados, la totalidad del circuito puede encargarse a un fabricante que lo realice en un sólo chip especial. Esta es una operación cara que sólo se justifica en el caso de grandes producciones. En el Spectrum se ha actuado de este modo por varias razones. En primer lugar, porque se ha fabricado en gran cantidad y, en segundo lugar, porque se pretende vender a bajo precio. Confiando una gran parte de la electrónica necesaria a la ULA se ahorra bastante dinero a largo plazo, sobre todo en el capítulo de montaje. Es, además, mucho más fiable el montaje de un solo chip que varios interconectados que trabajen satisfactoriamente. Cuando se fabrica una ULA puede ser verificada en la totalidad de sus funciones por medio de un ordenador, lo que significa un gran ahorro en el esfuerzo de montaje y pruebas finales.

De entre las varias funciones que realiza la ULA en el Spectrum destacamos algunas a continuación.

Decodificación de direcciones. La ULA realiza la tarea esencial de activación del bloque de memoria que la CPU está constantemente demandando. Si la CPU desea contactar, por ejemplo, con la memoria de vídeo para colocar alguna cosa en la pantalla, primero coloca la dirección de la memoria de vídeo en el bus de direcciones en la forma de un número binario de 16 bits. Esta palabra particular, compuesta de niveles lógicos de +5 y Ø V activa circuitos internos de la ULA.

Como resultado, uno de los terminales se coloca a nivel Ø. Dicho terminal está conectado al banco de memoria que contiene la memoria para la pantalla, y lo activa electrónicamente. Una vez activo, el banco de memoria utiliza su propio código interno de dirección para deducir, de la palabra que hay en el bus de direcciones, qué lugar en la matriz de memoria debe ser activado para recibir el dato. Una vez

que la dirección ha sido colocada por la CPU en el bus de direcciones, ésta coloca el dato de que se trate en el bus de datos para ser almacenado en dicha posición de memoria. La línea RD se colocará además a nivel alto, y la WR a nivel bajo debido a esta transacción, para significar que se ha producido una 'escritura' procedente de la CPU en la memoria.

Reloj. Como se puede ver, es importantísimo asegurar que las actividades del ordenador ocurran secuencialmente de una manera organizada. Las operaciones del ordenador deben llevarse a cabo de acuerdo con algún temporizador o *reloj*. La ULA es la encargada de suministrar este reloj a la CPU. El reloj que la ULA proporciona al Z8Ø consiste en un oscilador que bate tres millones y medio de veces por segundo, es decir, con una frecuencia de 3,5 megahercios (3,5 MHz).

También se encarga la ULA de proporcionar otro tipo de relojes para fines variados.

I/O (entrada/salida). La ULA es responsable de las líneas de entradas y salidas que posee el Spectrum. Cuando la CPU indica, por ejemplo, que debe activarse el zumbador, lo que hace simplemente es utilizar los buses de datos y direcciones de la misma manera que se indicó anteriormente, siendo la ULA la encargada de decodificar esta información electrónica y de activar la conexión del zumbador. Este mismo procedimiento se utiliza para la activación de la señal para la cassette o para la lectura del teclado. Veremos en el Capítulo 2 cómo puede aprovecharse esta manera de proceder, para controlar un dispositivo externo muy simple, tal como un relé o una luz eléctrica, utilizando una línea I/O de la ULA. Esto proporcionará al lector un excelente primer ejemplo de una situación de control.

Vídeo. La ULA controla la pantalla utilizando la información que la CPU y el programa han almacenado en la dirección de memoria correspondiente al vídeo. La pantalla es, efectivamente, una especie de 'ventana' elaborada, colocada en esta sección de la memoria. Cuando la CPU ha sido preparada para escribir la letra A, por ejemplo, en la pantalla, todo lo que la CPU tiene que hacer es escribir el código ASCII correspondiente a la letra A en la posición correspondiente de la vídeo-memoria. La ULA se encargará de interpretar que la memoria contiene los símbolos correspondientes al carácter que llamamos A y lo colocará en el lugar adecuado en el tubo de rayos catódicos del televisor. La posición exacta en que se

colocará la letra A dependerá del lugar que ocupaba en la vídeo-memoria. En la vídeo-memoria también se almacena la información cromática de cada carácter, y la ULA es la responsable de enviar esta información de color al televisor, conformando las señales convenientemente. Si estas operaciones fuesen realizadas en su totalidad por la CPU, sin ayuda, éstas serían lentas y tediosas. En efecto, el Sinclair ZX8Ø y el ZX81 no tienen un controlador especial para la pantalla, y es la CPU la que tiene que realizar todo el trabajo en todo momento. Esta es la razón, por ejemplo, de que la pantalla del ZX81 se borre cuando se presiona una tecla: la CPU ZX8Ø está controlando el vídeo y el teclado y tiene que abandonar la pantalla al servicio del teclado. Este no es el caso del Spectrum, ni tampoco el de la mayoría de los otros ordenadores.

La ULA, pues, alivia a la CPU de una gran cantidad de tareas mecánicas que podrían interferir con su capacidad para procesar el programa. Este mismo tipo de control puede encontrarse en otros ordenadores, pero raramente toda esta circuitería estará confinada en un espacio tan pequeño y económico como es la ULA del Spectrum. Gracias a este circuito integrado especial, puede construirse una máquina de la complejidad del Spectrum en tan pequeño tamaño y a tan bajo precio.

Miscelánea

Hay algunos otros bloques en el diagrama mostrado en la figura 1.2 que describiremos brevemente a continuación.

La alimentación en el Spectrum consta de dos bloques separados y conectados a través de un cable. El primer bloque (Alimentador ZX) contiene un transformador, rectificadores y un condensador de filtro, dentro de una caja propia. Convierte los 220 V de tensión alterna de la red eléctrica en 9 V continua. Puede suministrar hasta 1,4 amperios de corriente, según consta en la etiqueta de dicho alimentador.

El segundo bloque, PSU (fuente de alimentación), está localizado dentro de la caja del Spectrum y se encarga de convertir los 9 V en una serie de tensiones de diferentes valores, +5, +12 y —5 Voltios. No existen —12 V, como se relaciona equivocadamente en el diagrama de terminales exteriores en el manual del Spectrum. Dicho terminal contiene una tensión sin regular de +12 V.

La tensión más importante generada es la de +5V, por ser la tensión normal de alimentación de los circuitos integrados lógicos, y

también es una tensión muy utilizada por otros circuitos exteriores. La capacidad de suministro de corriente de la tensión de +5 V está muy aprovechada en la versión de 48K de la máquina, por lo que no debe utilizarse para alimentar circuitos exteriores. La versión de 16K puede proporcionar algo de corriente al exterior, pero no en gran cantidad. En el Capítulo 2 se describe una fuente de alimentación separada para alimentar circuitos exteriores.

La figura 1.2 también muestra un bloque RESET (reposición), conectado a la CPU. El Reset es un terminal eléctrico de la CPU que origina el comienzo de ejecución a partir de un área específica de memoria cuando se enciende la máquina. Cuando los circuitos electrónicos de cierta complejidad son alimentados, su estado interno inicial puede adoptar aleatoriamente una disposición cualquiera. No podemos estar seguros de que la CPU empiece a ejecutar instrucciones desde el principio del programa de control residente en la ROM, al encender la máquina. Puede encontrar alguna ramificación y ejecutar un lazo cerrado infinito. Para evitarlo, todas las CPUs poseen un terminal RESET que normalmente está a nivel 'bajo' (Ø, por ejemplo) tan pronto como se enciende, para forzar a la CPU a tomar un estado conocido y controlable. Como sería incómodo y a veces imposible esperar que el usuario apretara un botón cada vez que encienda la máquina, se realiza un circuito sencillo de reset automático inicial en todos los sistemas de ordenadores. El circuito reset mantiene el reset bajo durante unos momentos al encenderse el equipo, para dar tiempo a que las tensiones de alimentación se estabilicen, para volver a ser alto y comenzar desde este momento la ejecución normalmente.

El Z8Ø comienza la ejecución desde la dirección de memoria más baja tan pronto como el reset se hace alto, cuando la fuente de alimentación ya se ha estabilizado. Por ello, la ROM, que debe capturar la atención de la CPU inmediatamente, se coloca ocupando los primeros 16K de las direcciones de memoria, siendo automáticamente la primera cosa que la CPU 'mira' nada más efectuarse el reset.

Otro componente importante del Spectrum es su conector de expansión. Este permite al usuario tener acceso a la totalidad de las líneas de bus de la CPU y de otras importantes señales muy utilizadas. Por medio de interfaces construidos para este conector se pueden expandir las funciones del Spectrum de la manera que se desee. Este libro está dedicado a los métodos de utilización de este conector de expansión para variadas aplicaciones de control y de robótica. Puede usarse este conector para añadir al Spectrum un controlador

de disco, bancos de memoria extra, pantallas de vídeo auxiliares, etc. Desde luego, el software necesario para el control de estos dispositivos es tan complejo que se necesita un sistema grande de ordenador para su desarrollo. En este libro, sin embargo, se describen aplicaciones que sólo necesitan circuitos electrónicos auxiliares muy simples.

Mapa de memoria

Otro concepto importante es el *mapa de memoria* o lista de direcciones, que ocupa gran parte del contenido del ordenador. Un entendimiento del mapa de memoria de la máquina permite expandirla en un futuro sin interferir con los dispositivos actuales.

El manual del Spectrum nos informa de un detallado mapa de memoria de la máquina, pero está más orientado hacia el software que hacia el hardware. La tabla 1.1 muestra un mapa menos detallado, pero más orientado a las necesidades del diseñador para un entendimiento mejor de la máquina.

El mapa muestra sólo la memoria, pero la CPU Z8Ø tiene también un mapa separado, de direcciones de entrada/salida I/O, que describiremos posteriormente y que permite añadir dispositivos I/O al Spectrum. Es necesario que los dispositivos conectados a los buses distingan entre esos dos mapas, y para ello existen dos líneas de control de salida de la CPU para que estos dispositivos actúen correctamente. Estas dos líneas de control son las líneas IORQ y MREQ que significan 'petición de I/O' y 'petición de memoria', respectivamente. Esto se parece mucho al cometido de las dos líneas RD y WR que ya hemos visto. Cuando se aplica a un dispositivo de memoria una dirección a través del bus de direcciones, la línea MREQ se pondrá en el estado lógico bajo y la IORQ en el alto. Del mismo modo, cuando se trata de una dirección del mapa I/O, la IORQ se pondrá baja y la MREQ alta. La comprensión del comportamiento de las líneas de control RD, WR, IORQ y MREQ, más la utilización de los buses de datos y direcciones, es todo lo que se requiere para las aplicaciones de expansión, control y robótica. Más adelante veremos cómo deben utilizarse estas líneas.

El mapa de memoria está organizado mediante números expresados en decimal y hexadecimal. Para una mejor comprensión de ésto deberán consultarse el manual del Spectrum y el Apéndice 1 del final de este libro. Recuérdese que los números hexadecimales proporcionan un método bastante directo de observar la palabra binaria

Tabla 1.1. Mapa de memoria para el Spectrum

Banda de direcciones Hex	(Decimal)	K de almacenamiento y tipo	Utilización
FFFF	(65535)	32K RAM	Disponible para programación BASIC y almacenamiento de variables.
8ØØØ	(32768)		
7FFF	(32767)	(fin de la versión de 16K) 9,25K RAM	Disponible para programación BASIC y almacenamiento de variables.
5BØØ	(23296)		
5AFF	(23295)	6,75K RAM	Este bloque de RAM se dedica al vídeo.
4ØØØ	(16384)		
3FFF	(16383)	16K ROM	Este bloque contiene el sistema operativo
ØØØØ	(Ø)		

que está contenida en las líneas eléctricas del bus de direcciones, para activar los diferentes dispositivos de memoria del ordenador. Las líneas del bus de direcciones están numeradas desde la A15 hasta la AØ. La dirección expresada en hexadecimal nos permite obtener una idea exacta del contenido de unos y ceros en dichas líneas. Veamos, como ejemplo a qué corresponde la dirección de memoria expresada en hexadecimal como A5BC, utilizando para ello la tabla del Apéndice 1. La palabra binaria sería:

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	0	1	0	1	1	0	1	1	1	1	0	0
A				5				B				C			

Este conjunto de ceros y unos aparecería en el bus de direcciones, desde la CPU, en el preciso momento en que deba ser tomado por el decodificador de direcciones de la memoria, de tal modo que los datos sean escritos o leídos en la posición de dirección A5BC. La temporización de este suceso está encomendada a la CPU por medio de las líneas de control apropiadas. Veremos en un capítulo posterior cómo se realiza. Nótese que hay 16 líneas de direcciones, proporcionando un total de 2 elevado a 16 palabras binarias posibles, es decir, 65,536. Por ello se dice que el ordenador posee 64K bytes de memoria.

Mapa I/O

Para el lector de este libro, más importante que el mapa de memoria, es la estructura I/O de la CPU. El Z8Ø tiene 256 localizaciones de memoria de 8 bits llamadas *puertos I/O*, independientes de las del mapa de memoria de la Tabla 1.1. Estos puertos I/O son alcanzados por un conjunto especial de instrucciones de programa. Para leer y escribir en los puertos I/O se utilizan las palabras BASIC IN y OUT, de la misma manera que las PEEK y POKE se utilizan para leer y escribir en posiciones de memoria. Cada puerto tiene una dirección, éstas aparecen en las líneas normales de los buses de datos y direcciones. Sin embargo, cuando se está usando un puerto I/O, la línea IORQ se encontrará a bajo nivel, en lugar de la MREQ, como se dijo antes. Esta peculiaridad puede utilizarse para activar dispositivos externos. La Tabla 1.2 proporciona los estados lógicos en las cuatro líneas de control. Recordemos que estos estados lógicos son niveles de voltaje que pueden activar directamente dispositivos electrónicos, como veremos en breve.

Tabla 1.2. Estados lógicos de las cuatro líneas de control principales

MREQ	IORQ	RD	WR	*Acción*
Ø	1	Ø	1	La memoria es leída
Ø	1	1	Ø	Se escribe en memoria
1	Ø	Ø	1	El I/O es leído
1	Ø	1	Ø	Se escribe en I/O

Estos son todos los estados lógicos posibles en esas líneas. No se puede, por ejemplo, tener las dos líneas MREQ e IORQ a nivel bajo

simultáneamente, pues la memoria y los dispositivos I/O tratarían de utilizar el bus de datos al mismo tiempo.

Los 256 puertos I/O del Z8Ø son decodificados utilizando las 8 líneas inferiores del bus (A7-AØ), del mismo modo que se decodifican en las localizaciones de memoria. Sin embargo, el Spectrum no utiliza los 256 posibles puertos del Z8Ø muy eficientemente, ya que sólo emplea las líneas AØ-A4, dejando las A5, A6 y A7 libres para decodificación de direcciones exteriores. Esto sólo permite seleccionar ocho puertos distintos. En cambio, como cada localización de puerto o de memoria tiene ocho bits, uno por cada línea del bus de datos, hay $8 \times 8 = 64$ bits de I/O para utilización externa. Cada uno de esos 64 bits pueden comandar un interruptor o permitir colocarlo en un estado particular, si se usan como entrada, lo que es otro medio que puede utilizarse par aplicaciones robóticas y de control.

Conclusión

Este capítulo, con el complemento de los dos primeros apéndices, contienen las suficientes ideas para que se comprendan y puedan realizarse las prácticas y proyectos que se describen en el resto del libro. Mucha de esta información es tan general que permite comprender, además, la operativa de otros microordenadores y de la electrónica necesaria para otros sistemas, en general.

El control electrónico, como veremos en breve, es una aplicación idónea de las posibilidades de la CPU que puede realizar enviando señales electrónicas hacia el mundo exterior, a través de sus buses y puertos I/O. Estas señales pueden ser controladas a su vez por programas muy simples, que dependerán, así como su conexión, del propio diseño de la CPU y del sistema de ordenador del que forma parte.

2
Un ejemplo de control simple

Introducción

Este capítulo explica cómo puede usarse el Spectrum para controlar un dispositivo electrónico simple. Se utiliza una cantidad mínima de componentes electrónicos extras, al emplearse en el proyecto la electrónica existente en el Spectrum para decodificar un puerto de salida I/O (entrada/salida). Esto proporcionará al usuario una buena preparación para el tipo de programación y construcción electrónica que se requiere en este campo. Sin embargo, aquél tendrá que abrir la caja del Spectrum y realizar una simple conexión en un punto de su interior. No hay más remedio, pero si el Spectrum es nuevo y no se desea invalidar la garantía a causa de la apertura de la caja, no habrá más remedio que contentarse con la lectura cuidadosa de este capítulo cuya misión es presentar un importante número de conceptos usados en el resto del libro.

Asumimos que se ha leído el Apéndice 2, y también que ya se conoce cómo se construyen los dispositivos electrónicos, o que se ha experimentado con los ejemplos de ese apéndice. El Apéndice 4 proporciona algunos suministradores para ayudar a localizar los componentes electrónicos que se necesitarán.

El experimento descrito aquí llevará a controlar un dispositivo electrónico externo, pero sin usar otra entrada que el teclado. Podrá usarse esto, por ejemplo, para encender los motores y luces de un tren eléctrico, o para controlar una caldera de calefacción central, o su bomba, de acuerdo con el momento del día —el Spectrum puede servir como reloj 'en tiempo real'. Del mismo modo, pueden encenderse luces de una casa desocupada, por motivos de seguridad. Sin embargo, no se describe aquí el control de dispositivos más importantes, y no se recomiendan si no se posee un grado considerable de experiencia en circuitos eléctricos.

Este tipo de control no es nada complicado y normalmente no se

necesitaría usar un dispositivo tan complejo como un Spectrum para llevarlo a cabo. No obstante, de esta manera, se introducen algunos conceptos valiosos y que proporcionan una introducción simple a la teoría de control.

El siguiente paso, después de este ejemplo, es llevar alguna entrada automática a la máquina que haga posible el control para actuar de acuerdo con algunos estímulos exteriores. La entrada debería conectarse a través de los circuitos internos del Spectrum, pero para ello habría que levantar la tapa y soldar más hilos en el interior. El siguiente capítulo explica cómo expandir el Spectrum para entrada y salida, usando el conector de la ranura posterior y muestra cómo ciertos circuitos electrónicos pueden proporcionar muchas líneas de control en paralelo al Spectrum.

Durante la realización de proyectos en este capítulo y en el resto del libro, hay una o dos reglas importantes a observar. Por ejemplo, nunca debe conectarse nada a ningún circuito electrónico cuando esté bajo tensión. Hay que ser muy cuidadoso para no cruzar inadvertidamente hilos y patillas juntos mientras se estén verificando señales de algún tipo particular en un dispositivo alimentado. La concentración habrá de ser continua mientras se estén manipulando circuitos electrónicos, o se destruirá con gran facilidad. Si se toman estas precauciones, y se permanece continuamente atento, se disfrutarán muchas horas de un provechoso interés por la electrónica...

Finalmente, no debería comenzarse ningún trabajo hasta que no se haya acabado de leer el capítulo y se esté seguro de haberlo entendido suficientemente.

I/O (entrada/salida) en el Spectrum

La Figura 2.1 ilustra una pequeña parte del esquema eléctrico del Spectrum. Los componentes están numerados o descritos de acuerdo con los convenios usados en el Spectrum; si se necesita una explicación de los símbolos empleados, puede ayudar la lectura del Apéndice 2. Este diagrama muestra de qué manera están conectados a la ULA el zumbador, y las líneas MIC y EAR. La ULA gobierna los dispositivos I/O (entrada/salida) del Spectrum; veremos de qué manera realiza esta misión usando de manera hábil una sola línea de control. Como veremos, la línea (patilla 28 de la ULA) está conectada al puerto I/O 254 (decimal).

Todos los puertos I/O tienen 8 bits, y el MPU asume que cada bit está relacionado con la línea de datos (Bus de datos). Examine-

mos el puerto 254. La ULA reconoce esta dirección cuando aparece en la dirección del bus y toma el byte de 8 bits. Cada bit de dicha palabra (byte) lo utiliza la ULA para activar o desactivar un dispositivo particular dentro del Spectrum. Esto significa que el puerto 254 no está disponible para que lo utilicen circuitos externos. La tabla siguiente muestra el significado de cada bit en el puerto 254:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT Ø
—	EAR	—	zumbador	MIC		color del borde	

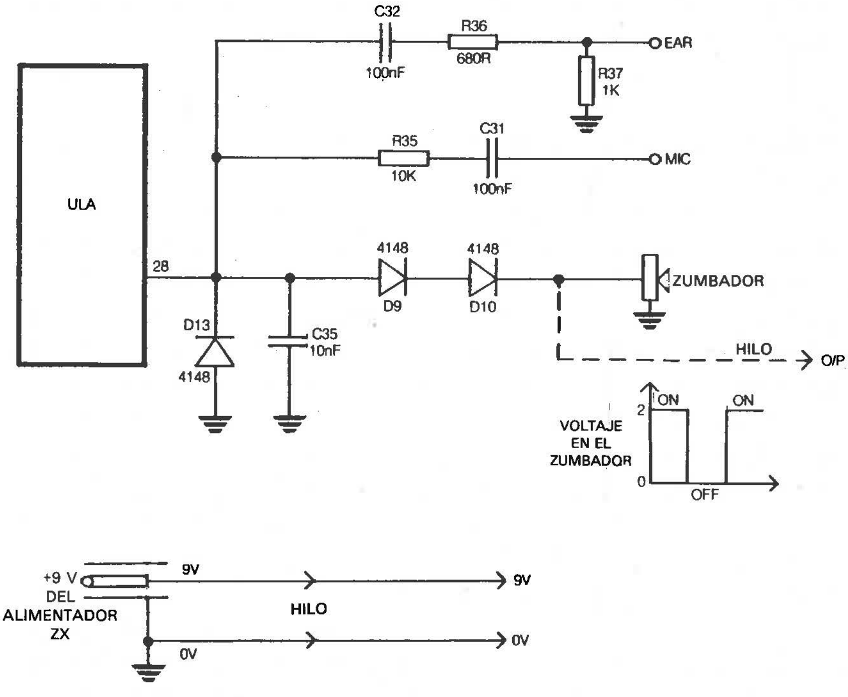


Fig. 2.1. Manejo de la ULA.

Si el bit 4 es 1 en el byte enviado a lo largo del bus de datos, el zumbador se activa. Puede ser desactivado de nuevo enviando cualquier byte que tenga el bit 4 a cero. El MIC se activa y desactiva del mismo modo usando el bit 3. El color del borde se puede cambiar por medio de combinaciones de los bits Ø, 1 y 2. EAR es, en realidad, una entrada y no se ve afectada por el byte enviado de esta forma. Pero si se teclea una sentencia de entrada IN 254, el estado del bit 6

lo lee el MPU y así permite inyectar a través de esta entrada la señal de salida de auricular de un magnetófono a cassette.

En la Figura 2.1 puede verse que la patilla 28 de la ULA se usa para tres dispositivos diferentes. Es el conjunto de bits que componen el byte enviado al puerto 254 el que determina el estado de esos dispositivos. Por ejemplo, como se explicará más adelante, la sentencia:

```
OUT 254,16
```

pondría la salida del zumbador a nivel alto, pues 16 es el valor decimal del número binario ØØØ1ØØØØ. Todos los bits, excepto el correspondiente al zumbador, son bajos. Además, por ejemplo, el color del borde debería haber cambiado a negro.

Manejo de las líneas de salida

Si se observa la figura 2.1, se verá que tanto EAR como MIC y el zumbador vienen todos de la patilla 28 de la ULA. ¿Cómo distingue la máquina entre ellos?

La patilla 28 de la ULA es electrónicamente una entrada y una salida a la vez. Está controlada por la ULA dependiendo del estado de RD y WR, que son activados por turno usando las sentencias IN o OUT, respectivamente. Por tanto, EAR no es operativo cuando se usa OUT, y los otros, cuando se usa IN. Pero puede haber interferencias mutuas. Usted ya debe conocer que no se puede grabar un programa en cassette cuando ambas clavijas EAR y MIC están conectadas. La línea MIC transporta las señales desde el Spectrum hacia el magnetófono cuando se graban programas en cinta. La línea EAR envía las señales desde la cinta hacia el ordenador para cargar programas grabados. Pero, cuando se graban programas, la mayoría de los magnetófonos reenvían parte de la señal que les llega a través de la línea MIC hacia atrás por la línea EAR. Esta es la causa de que se produzcan interferencias. La carga (LOAD), en cambio, no se ve afectada, pues la conexión MIC de los magnetófonos está inactiva durante la reproducción.

La ULA distingue entre las salidas de zumbador y MIC de una manera muy ingeniosa. Cuando la salida MIC se activa o desactiva, de acuerdo con el bit 3, la ULA proporciona un voltaje de Ø V, para el cero lógico, o 1,4 V para el uno. La señal de salida que se emplea en el conector MIC es una señal alterna producida por las conmutaciones sucesivas del estado del bit 3 del cero al uno. Estas oscilacio-

nes de corriente alterna son capaces de pasar a través del condensador C31, mientras que los voltajes de continua quedan bloqueados. La señal oscilante que necesita la entrada MIC de los magnetófonos es muy pequeña, y estos valores comprendidos entre Ø y 1,4 V son suficientes. Sin embargo, este pequeño voltaje no basta para pasar a través de los diodos D9 y D1Ø para activar el zumbador, y este dispositivo permanecerá desactivado.

Cuando el bit del zumbador (bit 4) es uno, la ULA proporciona un voltaje mayor, alrededor de 3,3 V, que sí pasa a través de los diodos y llega al zumbador con un voltaje de casi 2 V. Esto explica por qué el zumbador suena con poco volumen. 2 V es algo bajo; si este valor fuese elevado hasta 5 V, el zumbador sería significativamente más sonoro. Esto se puede hacer, pero habría que cortar una pista en el circuito impreso del Spectrum e insertar un transistor amplificador en serie con él.

Nótese que cuando el zumbador se le está enviando una tensión alterna, ésta aparece también en MIC. No hay medio de evitarlo.

Todo esto explica cómo podemos manejar una sola línea de salida, simplemente conectándola a la patilla 28 de la ULA. Poniendo el bit 4 del puerto 254 alto, cambiará la línea a activa (3,3 V), y reponiéndolo (bajo), volverá a ser inactiva. El próximo apartado mostrará cómo usar la señal del zumbador como una salida, y cómo controlar un relé desde ella.

La línea de salida

Como puede verse en la figura 2.1, hay un problema en el uso de los conectores EAR o MIC para conectar dispositivos externos a la patilla 28. Hay dos componentes en el camino. La resistencia no es tan importante, tan sólo atenúa la corriente disponible, y sólo es necesaria una corriente pequeña para hacer conducir a un transistor o un circuito integrado y, por tanto, controlar una salida. El principal problema es el condensador (C32 o C31). Estos condensadores no dejan pasar tensiones continuas; a través de ellos sólo pasan los voltajes cambiantes, es decir, las señales alternas. Cuando la patilla 28 se activa, el voltaje en dicha patilla pasa a ser de bajo a alto, y este cambio puede detectarse al otro lado del condensador. Pero inmediatamente se volverá a la anterior situación de equilibrio de continua que había antes del cambio de estado, en los conectores EAR o MIC, y que dependen de los circuitos externos a ellos conectados. Los condensadores hacen que sea muy dificultoso conectar circuitos

externos fácilmente controlables a los conectores MIC o EAR, y por esta razón es preferible tomar un hilo de la propia patilla 28. A pesar de todo, como se explicó antes, la patilla 28 no proporcionará una tensión de +5 V cuando esté activada, y ello puede ser también un problema, pues este valor de tensión va a sernos necesario para los componentes electrónicos que vamos a usar.

La Figura 2.1 muestra que en el propio zumbador, después de D9 y D10, hay un potencial de 2 V que bascula desde el valor lógico 1 (activado) al ∅ (desactivado), producido por el bit 4. Esto es perfectamente válido para comandar el transistor que se describirá más adelante, y siempre será mejor que conectar nada a la propia patilla 28. Así pues, será aquí de donde tomemos nuestro hilo de salida. Es un punto particularmente accesible para realizar la soldadura y fácil de identificar, y, además, los diodos proporcionarán alguna protección para la ULA.

Además de la línea de salida, la figura 2.1 muestra una conexión para la fuente de alimentación. Los +9 V del interior del Spectrum son una fuente útil de potencia y la usaremos para alimentar nuestro circuito, después de pasar a través de un circuito apropiado.

Elevador del nivel de 2 V a + 5 V

La Figura 2.2 muestra un circuito para pasar las tensiones de 0 y 2 V del hilo del zumbador a +5 y ∅ V, respectivamente. El hilo del

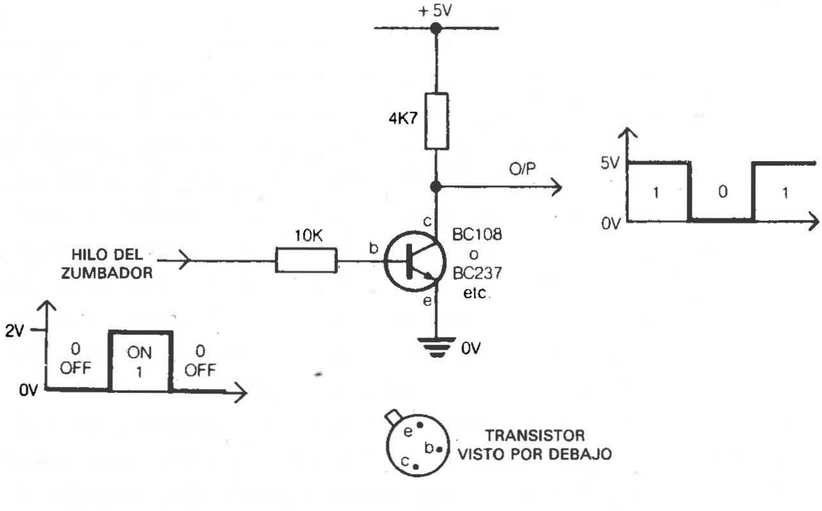


Fig. 2.2. Amplificación del CIP del zumbador.

zumbador está conectado a través de una resistencia (1∅ K) a la base (b) de un transistor. El transistor es simplemente un interruptor electrónico. Cuando a su base se le aplica un voltaje positivo pequeño, a través de la resistencia de 1∅ K. conduce y conecta su colector (c) a su emisor (e). Cuando esto ocurre, el hilo O/P se pondrá a un potencial de ∅ V, es decir, a nivel lógico ∅.

Cuando el transistor deja de conducir, es decir, si su base está a ∅ V, el O/P es elevado a +5 V por la resistencia de 4K7, y por ello toma el valor lógico 1. Nótese que ocurre lo inverso que a la entrada de la base. Cuando el hilo del zumbador está a ∅ V la salida es alta, y cuando el zumbador está a 1 (2 V) el O/P es bajo. Esto habrá de tenerse muy en cuenta a la hora de programar, ya que necesitaremos enviar un cero cuando queramos un uno a la salida de este transistor amplificador. El circuito mostrado forma parte del proyecto descrito más adelante. Usted podrá hacer que este O/P se active o desactive tan sólo pulsando cualquier tecla de su Spectrum. Cada vez que usted pulse una tecla el zumbador le proporcionará un "clik" audible. Este es el sonido de un impulso de 2 V que se aplica al zumbador, y el amplificador ahora lo convertirá en un impulso de nivel lógico de +5 V.

La construcción y el uso del circuito se describirá en breve. Para poder usarse necesitará una fuente de alimentación que proporcione la tensión de +5 V que aparece en el esquema. Esto se explica a continuación.

Una sencilla fuente de alimentación

El conector de la ranura posterior del Spectrum dispone de una patilla con +5 V para alimentación, pero en el Spectrum de 48 K esta alimentación está tan aprovechada que queda disponible muy poca energía sobrante. No es aconsejable, pues, su uso para proyectos. Sin embargo, si se posee un Spectrum de 16 K, puede alimentarse su circuito directamente desde los +5 V disponibles en el conector posterior, pero siempre será preferible usar para ello una fuente de alimentación separada.

La Figura 2.3 muestra una fuente de alimentación que utiliza un regulador de tensión (el circuito integrado IC-78∅5) para asegurar que los +5 V producidos son estables. Este es el mismo circuito integrado utilizado en el Spectrum, y se alimenta desde los +9 V que proceden del alimentador exterior ZX. No hay una gran cantidad de energía sobrante en este alimentador, pero basta para pequeños

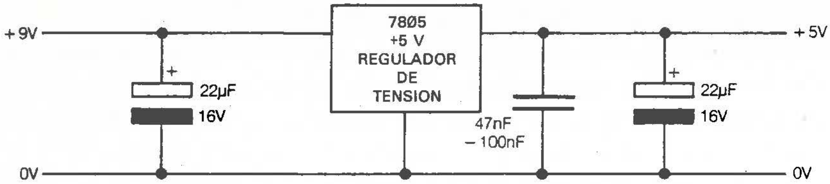


Fig. 2.3. Sencilla fuente de alimentación.

diseños. Más adelante veremos una fuente más capaz que uno puede construir.

Los otros componentes utilizados en la figura 2.3 son condensadores. Se necesitarán dos tipos de condensadores, electrolíticos (22 µF o más, a la tensión de trabajo mínima de 16 V), y un condensador cerámico de disco (cualquier valor comprendido entre 47 nF y 100 nF). Los condensadores electrolíticos funcionan como depósitos de energía para prevenir pérdidas de tensión cuando hay demanda de corriente, y el cerámico sirve para filtrar el ruido debido a componentes de muy alta frecuencia. Esto es muy importante cuando se alimenten circuitos integrados en los siguientes proyectos, y deberán añadirse otros discos cerámicos tan cerca como sea posible de las patillas de alimentación de cada circuito integrado, y con los terminales del mismo lo más cortos posibles.

Construcción de la fuente de alimentación

La Figura 2.4 muestra la construcción de la fuente de alimentación en una placa de pistas impresas standard. Esta placa tiene una matriz de agujeros espaciados en 0,1 pulgadas y líneas de cobre impresas a lo largo de la superficie inferior para soldar los componentes. La placa está dibujada por la parte superior, el lado aislado, y todos los terminales de los componentes se pasan por dichos agujeros para ser soldados a las pistas de cobre por el otro lado. Dichas pistas servirán para interconectar los distintos componentes entre sí. Algunas de ellas deberán cortarse para evitar demasiadas conexiones y posibles cortocircuitos. Los lugares donde el cobre debe ser cortado están indicados con cruces. Hay que procurar realizar los cortes en los agujeros donde la cantidad de cobre es menor, usando una broca de suficiente tamaño, o una herramienta de corte especial que uno pueda comprar. Cada corte deberá ser inspeccionado con una lupa para asegurarse de que se ha hecho bien.

Como se muestra, el regulador 7805 tiene tres terminales y una

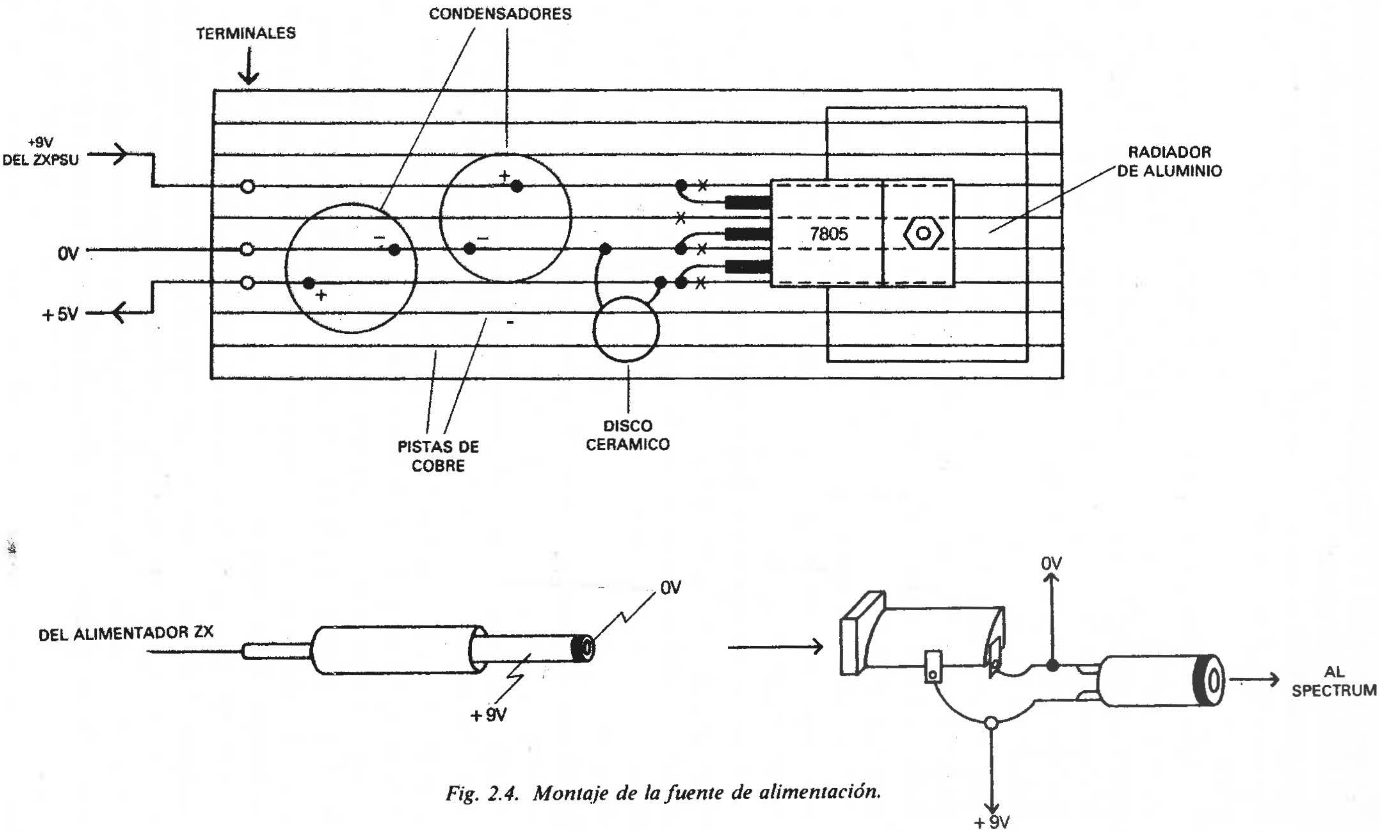


Fig. 2.4. Montaje de la fuente de alimentación.

parte metálica perforada con un agujero. Esta parte está conectada al terminal central, que a su vez está conectado a Ø V en la figura 2.4. La parte metálica es para disipar calor, y el regulador debe ser fijado a la placa. Para ayudar a la disipación de calor del regulador es una buena práctica el colocar una lámina de aluminio entre éste y la placa.

Los condensadores electrolíticos dibujados son de tipo radial. Si no se encuentra más que del tipo axial, también puede ser perfectamente aceptable. Asegúrese de que cualquier condensador electrolítico que emplee lo conecta con la popularidad adecuada. Los valores posibles de estos componentes pueden ser cualquiera que sea superior a 22 μF y un valor nominal de tensión de trabajo de 16 V como mínimo. Inténtese evitar los componentes demasiado grandes o incómodos, así el montaje será más limpio. El condensador cerámico no tiene polaridad y puede conectarse en cualquier sentido, pero hay que mantener los terminales cortos.

Los voltajes de entrada y salida deberán ser conectados a la placa por medio de terminales insertados en los agujeros indicados y correctamente soldados.

Ahora conéctese esta tarjeta al Spectrum por medio de los conectores que se indican en la figura 2.4. El alimentador externo ZX dispone de una clavija para conectar en el Spectrum y de ella pueden tomarse los hilos necesarios para sus propios conector y clavija. Procure usar el mismo tipo utilizado en el Spectrum y conéctelos de la forma que indica la figura. El alimentador ZX entonces deberá enchufarse a su conector y, a su vez, conectar su clavija a la entrada de alimentación del Spectrum. Asegúrese que su conexión mantiene la misma polaridad original que la de la clavija del alimentador, y procúrese de que no queden trozos de hilo desnudo muy largos que al tocarse dañen la fuente de alimentación.

Otro medio simple de conexión consiste en utilizar el conector de la ranura posterior del Spectrum, al cual se envían los +9 V procedente del alimentador ZX. Puede verse esto en el Apéndice 3, donde se describe dicho conector.

Cuando haya terminado la fuente de alimentación deberá inspeccionar todas las uniones y cada pista de cobre para verificar que no hay conexiones cruzadas entre conductores próximos. Después puede ser conectado el circuito a la tensión y comprobar, con la ayuda de un polímetro, que la tensión de salida es de +5 V.

Control de un relé

Para accionar un motor, luces, modelos, etc., es necesario hacer que se cierre algún interruptor. Es lo que se hace al pulsar la llave de la luz o el botón de encendido de la televisión. El ordenador puede producir voltajes lógicos que deberán ser convertidos en cierres de contactos eléctricos por medio de dispositivos exteriores. Un procedimiento normal de llevarlo a cabo es usando el relé eléctrico. El relé es un dispositivo que dispone de un interruptor mecánico accionado magnéticamente cuando se aplica una tensión a una bobina de hilo (un electroimán). Este apartado explica cómo podemos convertir cambios lógicos (+5 V y ∅ V) en cierres y aperturas de contactos eléctricos.

De nuevo se emplea un transistor para conseguirlo. Lo usamos, pues, para convertir los voltajes lógicos en corrientes que circulen por la bobina del relé. La Figura 2.5 muestra un circuito para hacerlo.

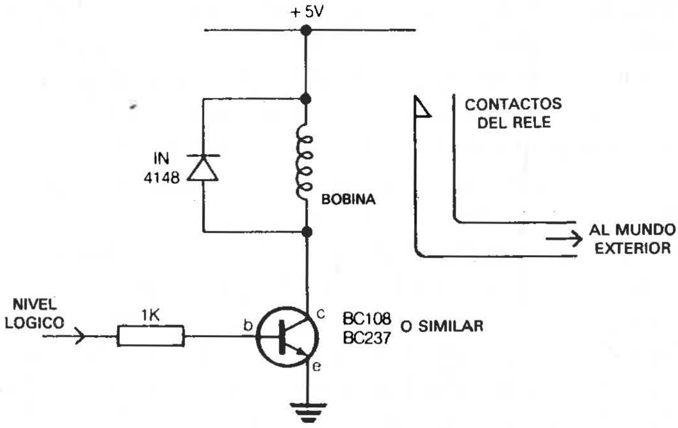


Fig. 2.5. Control del relé.

El transistor se hace conductor cuando se le aplica un nivel lógico 1 a la resistencia en serie de 1K con la base (b). De nuevo vemos que el transistor actúa como un interruptor que cuando se actúa conecta su colector (c) a su emisor (e) y ambos a ∅ V.

Esto hace que la corriente pueda circular por la bobina del relé y atraiga los contactos. El diodo (IN4148) es un diodo especial de protección. Cuando a una bobina por la que circula una corriente se le obliga bruscamente a que deje de hacerlo (abriendo el circuito que le proporcionaba la tensión necesaria para ello), se crea en la misma

un pico de tensión negativa de suficiente magnitud como para llegar a destruir el transistor que ha provocado dicha apertura. El diodo protege al transistor de este pico de tensión.

Los contactos del relé forman un simple interruptor controlado electrónicamente y son el enlace final de la cadena desde el "software" del Spectrum hasta el mundo exterior. Por medio de estos contactos usted podrá activar cualquier dispositivo que desee, desde pequeños aparatos alimentados con baterías a otros más potentes. Si se necesitaran manejar potencias mayores, siempre podrá añadirse otro relé capaz de manejarla. De esta manera puede amplificarse la potencia conmutada tanto como se necesite.

El próximo apartado muestra cómo realizar el circuito de salida completo.

Realización del circuito de salida

La Figura 2.6 muestra el esquema del circuito completo. Hay dos transistores, uno para cambiar de 2 a 5 V y el otro para cambiar ésta en corriente para el relé. Es posible combinar estas dos etapas en una sola, pero ello requeriría absorber más corriente del hilo del zumbador, y no se recomienda. El segundo transistor que maneja el relé es un buen circuito general que volveremos a usar para convertir niveles lógicos en corrientes de excitación de relés.

La Figura 2.6 también proporciona una completa explicación para el montaje. Como puede verse, hay seis componentes con sus terminales pasados a través de los agujeros de la placa, y soldados por la parte inferior de la misma. Nótese que el diodo de la figura 2.5 no se necesita si usamos el relé que está relacionado en la lista de componentes, pues está incluido dentro de la caja del relé. Es importante que el relé, por este motivo, sea colocado con la polaridad correcta, o el diodo cortocircuitaría la bobina.

La lista siguiente es una relación de componentes necesarios para la realización práctica del circuito.

Ctd.	*Componente*	*Marcas*	*Notas especiales*
1	Resistencia 1ØK	Mar/Neg/Nar	Resistencia 1/4 vatio.
1	Resistencia 1K	Mar/Neg/Roj	Resistencia 1/4 vatio.
1	Resistencia 4K7	Ama/Vio/Roj	Resistencia 1/4 vatio.
2	Transistores BC1Ø8/BC237	Ver Ap. 2	Cualquier transistor de conmutación NPN.

Ctd.	*Componente*	*Marcas*	*Notas especiales*
1	Relé 5ØØ ohm. 5 V con conector DIL		Un relé con un voltaje de trabajo mínimo del orden de 3,5-5,5 V. La resistencia no deberá ser mucho menor que 1ØØ ohms para obtener el mejor efecto. El que se muestra es del tipo RS 348-582.
1	Placa de pistas impresas		Veroboard con agujeros cada 0,1 pulgadas.
1	Caja de terminales		Terminales Vero para conexiones externas.
1	Hilo de conexión		Hilo de cobre estañado recubierto de plástico.

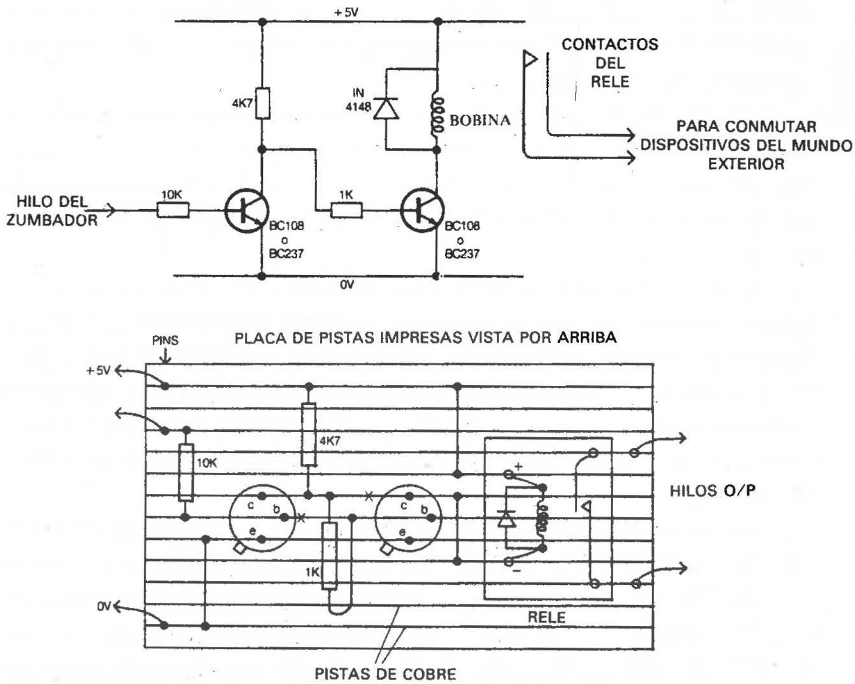


Fig. 2.6. Circuito de salida.

Todos estos componentes pueden conseguirse fácilmente y no es necesario romperse la cabeza buscando exactamente el componente físicamente igual al que hemos propuesto, aunque, eso sí, si se compra un componente alternativo, el vendedor deberá garantizarle que es equivalente. En caso de duda, si se considera conveniente, enséñese al suministrador el esquema del circuito y explíquese lo que realmente se desea.

Las pistas de cobre de la placa se usan fundamentalmente para soldar los componentes. Con dichas pistas podrán conseguirse la mayoría de las interconexiones del circuito. Cuando, a pesar de todo, éstas no son suficientes, se usan trozos de hilo aislados. Se debe utilizar hilo de cobre estañado aislado con plástico. Estos hilos se medirán colocados sobre la placa, se cortarán y pelarán para posteriormente introducirlos por las perforaciones y soldarlos como si se tratara de cualquiera de los componentes. No debe usarse una longitud excesiva, que dará un aspecto desordenado a la misma y tendrá dificultades para encontrar averías, si se producen. Asimismo, manténganse los terminales de los componentes tan cortos como se pueda para prevenir cortocircuitos no deseados. Es buena práctica el uso de macarrón de plástico para aislar los terminales. Todos los componentes deberán pasarse por los taladros desde la parte de la placa aislada hacia el lado de las pistas de cobre, y soldados. Elimínese el sobrante de los terminales cortándolo lo más cerca posible de la superficie de la placa.

Para el tipo de relé citado se consiguen también fácilmente bases o zócalos. Siempre es conveniente el uso de zócalos y componentes enchufables en los mismos, pues ello permite la reutilización de componentes con facilidad, y ayuda a reemplazarlos en caso de avería.

Al igual que para la fuente de alimentación descrita anteriormente, cuando las pistas de cobre impreso conectadas al circuito ocupen porciones no deseadas de la placa, deberán cortarse para evitar contactos. Los puntos de corte que se sugieren están marcados con cruces, pero los cortes deben comprobarse con una lupa y asegurarse que no quedan restos de cobre, ya que esto suele ser una causa de errores.

Los transistores empleados son del tipo de encapsulado metálico, con una pequeña orejeta en la base del encapsulado para su orientación. Si sólo se encuentran transistores con encapsulado de plástico, no olvide preguntar, cuando lo compre, por la posición de sus terminales, ya que su disposición puede variar considerablemente de uno a otro tipo.

No hay que preocuparse demasiado por ajustarse exactamente a la configuración mecánica adoptada en la figura. Utilícese como ejemplo para producir la propia estructura, dependiendo de dónde vaya a colocarse la unidad. Según sea de plástico o metal la caja que la contenga, se necesitarán tarjetas de distintas dimensiones. Lo realmente importante es que todas las conexiones del esquema estén correctamente realizadas, sin importar gran cosa el medio empleado para ello. También puede incorporarse la fuente de alimentación en la misma placa, si se desea.

Prueba del circuito

Antes de conectar el circuito al Spectrum deberá ser probado. Compruébese una vez más todo el cableado cuidadosamente. Verifíquese que no hay cortocircuitos o restos de estaño y, luego, conéctese a la fuente de alimentación ya comprobada.

Para ver si funciona, hay que conectar simplemente el extremo del hilo destinado al zumbador a ∅ V. Esto hará que se cierren los contactos del relé. Conectándolo a +5 V se separarán. Si no funciona, habrá que revisar todo el montaje desde el principio, comenzando por apagarlo.

Si se tiene un modelo o una linterna de pilas que controlar, se deben conectar al relé ya, y probar su funcionamiento antes de emplear el Spectrum.

Conexión al Spectrum

Para conectar el hilo del zumbador al circuito, hay que abrir la caja del Spectrum. Colóquelo sobre una superficie limpia y proceda a quitar los tornillos de la parte inferior. Vuelva a colocarlo sobre su base, pero tenga cuidado de que no se abra la caja cuando invierta su posición. Una vez colocado hacia arriba, separe un poco la parte superior. Se observará que dentro hay dos cables de cinta de pequeña longitud que conectan la parte superior (teclado) con la inferior (circuito impreso). Procure que estos cables no se estiren al separar las dos partes de la caja, pues están conectados en sus extremos a la tarjeta de circuito impreso (PCB) por medio de dos conectores que, preferiblemente no deben ser desconectados. Si, por cualquier causa, esto ocurriera, debe volverlos a colocar en su sitio sin retorcer el cable. No se olvide de dejarlos bien conectados.

Para tener acceso al punto donde hay que conectar el hilo coloque la caja con el frontal del ordenador hacia usted y levante y deslice ligeramente la parte superior de la caja (la correspondiente al teclado); podrá ver el zumbador en la esquina inferior derecha de la placa de circuito impreso, junto a la placa disipadora de calor del regulador de tensión. Comprobará, asimismo, que el zumbador tiene dos terminales soldados al circuito impreso (ver figura 2.7). El hilo deberá soldarse al terminal de señal del zumbador, no al de masa. En las tarjetas correspondientes al Spectrum Versión Dos, dicho terminal es el inferior. Pase a continuación el otro extremo del hilo por debajo del teclado hacia la ranura posterior del conector y sáquelo hacia fuera. Ahora tendrá que volver a cerrar la caja. Cuando lo haga tenga cuidado de que los cables no se retuerzan ni se dañen. Vuelva a colocar los tornillos y la operación quedará terminada.

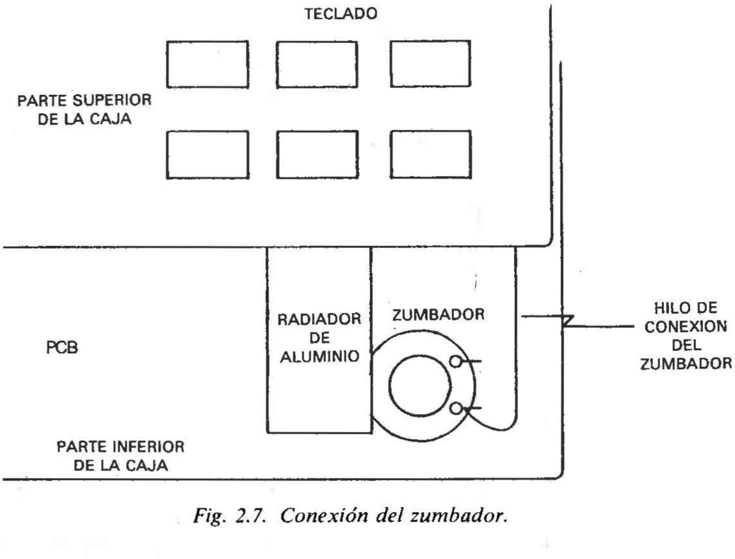


Fig. 2.7. Conexión del zumbador.

Los hilos de +9 y Ø V pueden también soldarse en el interior de la caja del Spectrum en el propio conector, si se desea, pero hay que realizar las soldaduras en el sitio correcto, de lo contrario podría dañar el circuito.

Utilización de la placa de salida

Una vez verificado el sistema manualmente deberá conectar ya el hilo del zumbador del Spectrum. Para ver el efecto del experimento,

conecte al relé el dispositivo que usó para la prueba manual, o bien utilice un diodo luminiscente (LED) en serie con una resistencia, como se muestra en la Figura 2.8, alimentado por la fuente de +5 V. Por supuesto, puede utilizar en su lugar cualquier bombilla de linterna o el dispositivo que desee controlar.

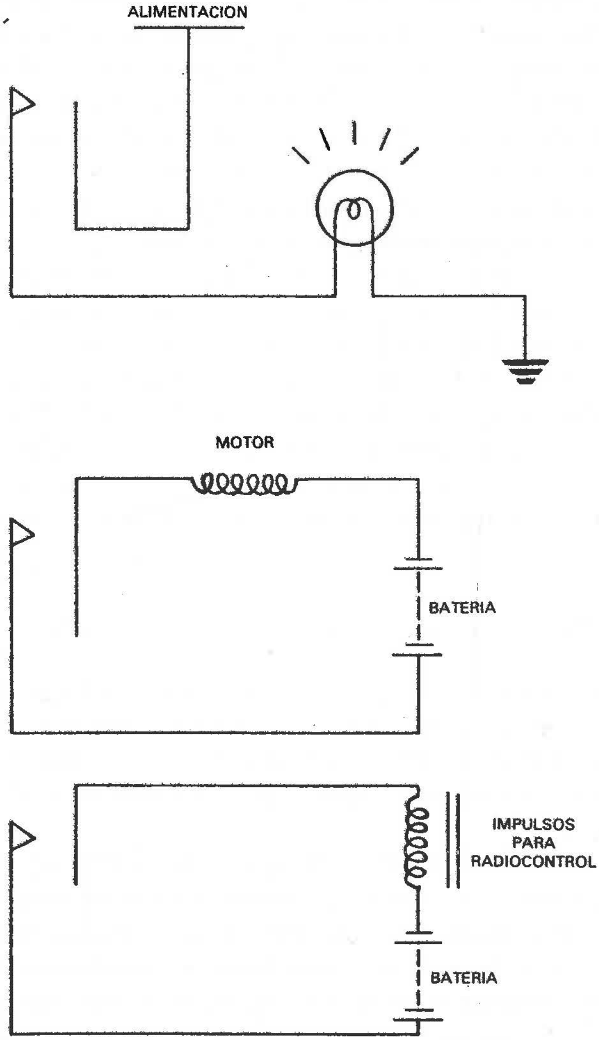


Fig. 2.8. Utilización del interruptor del relé.

Con el hilo del zumbador ya conectado podrá verificar el funcionamiento del relé usando las sentencias de programa del principio del capítulo. Intente ejecutar el siguiente programa:

```
10 OUT 254,255
20 PAUSE 50
30 OUT 254,239
40 PAUSE 50
50 GO TO 10
```

Este programa debe hacer el bit correspondiente al zumbador alto, después de un retardo, bajo; a continuación de otra pausa, de nuevo alto. Esto se repetirá indefinidamente a menos que se pulse el comando BREAK. Si el controlador no siguiese esta secuencia, es decir, si el relé no abre y cierra sus contactos continuamente, deberá apagarse el equipo y verificar todo el proceso desde el principio.

El programa demuestra cómo trabaja la tarjeta de salida, y proporciona la primera idea de cómo un ordenador puede conectar y controlar el mundo exterior. Desde luego pueden existir muchos programas que uno mismo puede escribir para este tipo de control. Por ejemplo, se puede combinar el teclado con un programa de reloj en tiempo real para controlar la señal luminosa de un tren eléctrico. El programa deberá comenzar preguntando a qué hora quiere que ésto ocurra, y almacenar esta instrucción. Cuando llegue la hora indicada, el programa hará que se cierren los contactos del relé y, por tanto, se encenderá lo que hayamos conectado a éstos.

Aplicaciones

El principal motivo de describir este sencillo controlador ha sido introducirle en la ocupación de convertir a su interruptor del Spectrum en un dispositivo del "mundo exterior". Hay muchas cosas que se pueden hacer con este circuito, y en este apartado mostramos un par de ejemplos.

En la Figura 2.8 podemos ver tres tipos de dispositivos que se pueden accionar con el relé. Se pueden encender y apagar luces eléctricas, en cualquier hora del día o de la noche, bajo control del ordenador. Esto sugiere varias aplicaciones. Se puede construir una luz de seguridad para los momentos en que esté fuera de casa. Bastará programar que la luz se encienda a la caída de la tarde hasta entrada la noche para engañar a posibles ladrones. Al mismo tiempo, puede conectarse una cassette grabada con sonidos especiales, para mayor seguridad.

Las señales luminosas pueden utilizarse también para otros objetivos. Por ejemplo, si se está interesado en la manera de reaccionar

de las personas ante estímulos, puede hacer que aprieten un tecla cuando se encienda una luz. Esta es la base de una experiencia interesante sobre la diferencia que existe entre los estímulos procedentes de la pantalla del televisor y de la luz simple. El individuo deberá responder cuando algo concreto aparezca en la pantalla y luego, de la misma forma, cuando se encienda una luz. Los tiempos de reacción pueden ser almacenados en la memoria del ordenador, así como el nombre del individuo, para poder imprimir posteriormente los resultados o confeccionar gráficos estadísticos con ellos.

Existen muchas variaciones sobre este tema, incluyendo, por ejemplo, la manera de reaccionar bajo la influencia de estímulos sonoros.

El relé deberá accionar entonces algún dispositivo sonoro al mismo tiempo que aparece algo en la pantalla. La persona responderá con una u otra acción, dependiendo de la combinación de que se trate de ambos estímulos. De nuevo, el ordenador, actuando como monitor, coleccionará todos estos datos automáticamente. Todo ello podría ser parte de un buen proyecto técnico escolar.

De la misma manera que el controlador actúa solamente sobre una línea, se puede extender el control a muchas más líneas, utilizando un "secuenciador". Este dispositivo lo utilizan los entusiastas del radiocontrol. Esencialmente, lo que hace es activar, de forma secuencial, distintas líneas por medios mecánicos cada vez que se le aplica un impulso eléctrico. Se utiliza mucho en los equipos de radiocontrol de canal único. Muchos secuenciadores son sencillos motores de relojería, con un electroimán, que mueven una leva de un lado a otro, consiguiendo hacer girar al dispositivo un cuarto de vuelta, por ejemplo, por cada impulso que se le envíe. De esta manera, conectando una palanquita convenientemente al eje del motor, se podrán abrir y cerrar microinterruptores para controlar diferentes cosas.

Desde luego, para alcanzar un interruptor, hay que haber alcanzado previamente los anteriores, pero ésto no es un inconveniente grave para muchas aplicaciones. Por ejemplo, utilizándolo para un sencillo sistema de seguridad, consistente en el encendido de las luces de una casa en varias habitaciones, de una manera secuencial, como si se estuviese cambiando de habitación.

Las aplicaciones para trenes eléctricos, o regulación de semáforos para el tráfico, son obvias. Sencillamente se trataría de encontrar la secuencia de acciones temporizadas para dar realismo al modelo. El mayor inconveniente de este controlador es que no existe realimentación del mecanismo que se está controlando. En el ejemplo

anterior del secuenciador la secuencia debe ser iniciada en un punto conocido, para lo cual el ordenador debe ser informado de la situación actual del dispositivo en cualquier momento. Otra circunstancia que puede ocurrir es que haya algún problema con los mecanismos; el ordenador no se entera y puede estar fuera de fase con el verdadero estado del sistema.

En el siguiente capítulo trataremos de los interfaces y periféricos que pueden añadirse al Spectrum para llevar a cabo la realimentación necesaria. También se tratará del diseño de nuestros propios circuitos periféricos, utilizando el conector situado en la ranura de la parte posterior del Spectrum.

3
Algunos interfaces reales

Introducción

Este capítulo se ocupa de los tipos de interfaces y periféricos que se necesitan para control y robótica. Muchos de los periféricos normalizados que se encuentran en el mercado para el Spectrum utilizan los principios que se describen a continuación. Seremos capaces de aprender lo suficiente para entender la mayoría de ellos e incluso diseñar los propios. Esto tiene varias ventajas. En primer lugar, podemos ahorrar dinero; en segundo lugar, seremos capaces de construir lo que necesitemos en cada momento y comprender lo que estamos haciendo. Si somos capaces de entender cómo trabaja un periférico podremos sacarle el máximo provecho a su utilización, encontrando posiblemente nuevas aplicaciones.

La primera experiencia que veremos consiste en la utilización de un chip I/O denominado PIO, de la familia del Z8Ø, que se conecta casi directamente en el Spectrum.

Este chip tiene 16 bits de entrada y salida y puede ser utilizado para, a través de él, interconectar con el Spectrum muchos tipos de sistemas de control y robótica. Veremos, a través de un sencillo ratón robot que podremos construir, cómo se controla por medio del PIO.

Conexión de periféricos

En el Capítulo 2 hemos visto cómo se puede controlar un dispositivo externo sin ningún tipo de realimentación del mundo exterior. Este capítulo nos proporcionará los medios de comunicación bidireccional, con el dispositivo a controlar, que son necesarios para que el ordenador no sólo efectúe las instrucciones de control, sino también reciba información del resultado de la misma, para, de esta

manera, realizar el control con precisión e inteligentemente. La primera tarea consiste en dotar al Spectrum de un medio de expansión que le permita disponer de varias líneas de entrada y salida. El Capítulo 1 nos mostró cómo algunas líneas de control y los buses de la CPU se utilizan para alertar variados dispositivos cuando son activadas. Ahora veremos cómo se puede generalizar al caso de periféricos externos, conectados a través del conector posterior de expansión. Se necesitarán, para ello, los buses de datos y direcciones, así como las líneas RD, WR e IORQ, por lo que sería conveniente efectuar un repaso de los conceptos ya adquiridos en el Capítulo 1.

Debe recordarse que sólo están disponibles para circuitos exteriores las líneas A5, A6 y A7. Para la activación de nuestros periféricos utilizaremos estas tres líneas de direccionamiento y las RD, WR e IORQ.

Un chip paralelo I/O

En este apartado se describe cómo se puede utilizar uno de los circuitos integrados más fáciles de conseguir en el mercado, para conseguir 16 bits de entrada y salida. Dicho CI se denomina "I/O paralelo" (PIO), pues todas las líneas I/O del chip están disponibles a la vez, en paralelo.

La mayoría de los módulos I/O que se venden en el mercado utilizan un solo chip I/O. Existen muchos chips especiales, y cada familia de CPU tiene normalmente varios de ellos. Están diseñados para conectarse directamente en los buses de datos, direcciones y control con el mínimo de equipo intermedio. La característica común a todos ellos es que tienen localizaciones de memoria interna que se utilizan como un medio para acceder a los bits I/O y controlar las actividades del propio chip. Estas memorias internas se denominan registros.

La Figura 3.1 muestra un diagrama de un chip de la familia Z8∅ denominado PIO (Parallel Input Output). En el Apéndice 2 se describen los métodos normalizados de representar los terminales de los circuitos integrados. Existen diferentes tipos de este chip, según la velocidad, que depende de la frecuencia de reloj del Z8∅. Para el Spectrum, se debe adquirir la versión A del PIO, que se denomina Z8∅A-PIO. El PIO es un chip complejo; para conocer la totalidad de las funciones que puede realizar hay que acudir a la información suministrada por el fabricante. Sin embargo, puesto que la utilización que vamos a hacer de él estará restringida a su conexión con el

Spectrum, nos limitaremos aquí a explicar las que nos resulten necesarias para conseguir realizar el control paralelo I/O.

La Figura 3.1 proporciona los terminales que vamos a necesitar utilizar; no muestra algunos que no nos serán necesarios para el presente propósito. Hay varias denominaciones familiares de los terminales del chip, así, las ocho líneas del bus de datos, las líneas de control IORQ y RD. Existen, además, otras líneas de control y tres líneas que conectaremos al bus de direcciones.

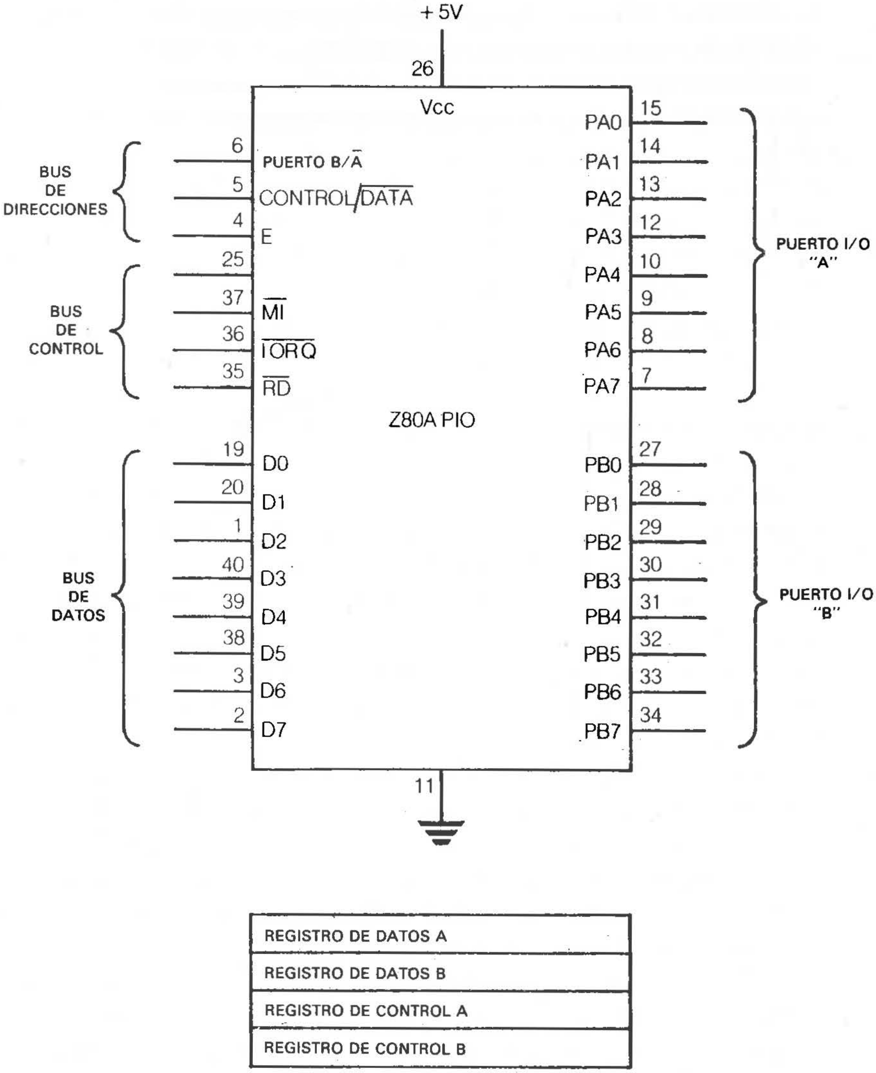


Fig. 3.1. PIO.

Hay también dos puertos I/O de 8 bits, llamados puerto A y puerto B. Estos terminales están etiquetados como los PAØ-PA7 y los PBØ-PB7, respectivamente, y pueden ser llevados a tomar estados lógicos para controlar relés, por ejemplo; o sea, para controlar dispositivos externos. Además, pueden ser utilizados para introducir estados de líneas de llegada y, de este modo, conseguir cerrar el lazo de realimentación.

Dentro del PIO hay cuatro registros, como se puede ver en la Figura 3.1. El bus de direcciones se utiliza para seleccionar las localizaciones de memoria externas, que se activan solamente cuando la línea IORQ está a nivel bajo, ya que éste es un dispositivo I/O. De aquí se deduce la necesidad de que el PIO tenga una conexión con la línea IORQ. Del mismo modo, la RD se conecta para informar al PIO en cada momento si ha de ser escrito o leído con los datos que entran o salen de los registros del PIO que están presentes en el bus de datos. Nótese que las líneas PA y PB están numeradas en el mismo orden que las líneas del bus de datos, de acuerdo con los datos que estarán en PA y PB. Así, por ejemplo, el PAØ y/o el PBØ estarán asociados siempre con el dato que ocupa la posición Ø (bit Ø).

Registros del PIO y decodificaciones de direcciones

Dos de los registros del PIO se llaman registros de control y se utilizan para llevar a cabo correctamente la operación del chip. Cada lado del PIO (lado A y lado B) tiene su propio registro de control. Antes de que el chip pueda ser utilizado, estos registros deben escribirse con algún byte de control. Los otros registros se denominan registros de datos, uno para cada lado, y los bits almacenados en los mismos se utilizan para llevar las líneas PA y PB a los estados lógicos que se necesiten.

Una característica del PIO es el hecho de que los registros de control pueden ser utilizados para colocar cada línea individual PA y PB, bien como entrada o como salida. Se pueden necesitar, por ejemplo, 9 entradas para una determinada aplicación y tan sólo dos salidas. En este caso, se podría definir un puerto como todo de entrada, y el otro con una línea de entrada y el resto de salida.

Para poder escribir bytes en los registros del PIO, el bus de direcciones debe ser decodificado para permitir al PIO acceder a algunas direcciones del mapa I/O. Esto significa que cuando escribimos una sentencia OUT, el dato contenido en ella debe entrar electrónicamente al PIO cuando el bus de direcciones y el bus de

control están controlados por la CPU para enviar los estados correctos. El PIO lo recoge de las líneas del bus de control y lo decodifica automáticamente, estando el bus de direcciones adecuadamente dispuesto para direccionar al chip en la forma correcta.

Para clarificar un poco todo esto, en la figura 3.1 se muestran los terminales 5 y 6, a los que nos vamos a referir para describir su funcionamiento. Cuando el terminal 6 está alto, por ejemplo, quiere decir que el lado B del PIO está activado, y cuando está bajo se activa el lado A. Cuando el terminal 5 está alto, uno de los registros de control está siendo direccionado, y cuando está bajo, será un registro de datos el direccionado. Mediante las líneas del bus de direcciones se pueden direccionar directamente los cuatro registros, como puede verse en la Tabla 3.1.

Como ocurría en los bloques de memoria descritos en el Capítulo 1, en el PIO también existe un terminal E-barra, que debe estar bajo cuando el chip está activo. Esta línea puede ser conectada a una línea del bus de direcciones, completando así la decodificación de direcciones del PIO. La figura 3.3 muestra cómo se interconecta el PIO con el Spectrum. Todas las líneas de la izquierda están disponibles en el conector de expansión del Spectrum. En breve veremos cómo se conectan.

Como se puede ver, A7 se conecta al terminal E-barra, que debe ser cero para que el PIO sea activado. La tabla de la figura 3.2 también muestra los estados de A5 y A6 que sirven para seleccionar

Tabla 3.1. Decodificación de direcciones del PIO

Direcciones									
Binario								*Decimal*	*Registro/acción*
A7	A6	A5	A4	A3	A2	A1	AØ		
Ø	Ø	Ø	1	1	1	1	1	31	Dato hacia y desde el Puerto A
Ø	Ø	1	1	1	1	1	1	63	Dato hacia y desde el Puerto B
Ø	1	Ø	1	1	1	1	1	95	Control Registro A
Ø	1	1	1	1	1	1	1	127	Control Registro B

los diferentes registros de acuerdo con la manera en que las líneas del bus de direcciones están conectadas al PIO. Hay que recordar también que una peculiaridad del Spectrum es que las líneas A4-AØ

deben estar en estado 1 cuando se realizan direccionamientos hacia cualquier circuito externo. La tabla 3.1 proporciona las direcciones a las que hay que dirigirse para acceder a los registros del PIO. Usaremos esta tabla posteriormente, cuando realicemos programas de control del PIO.

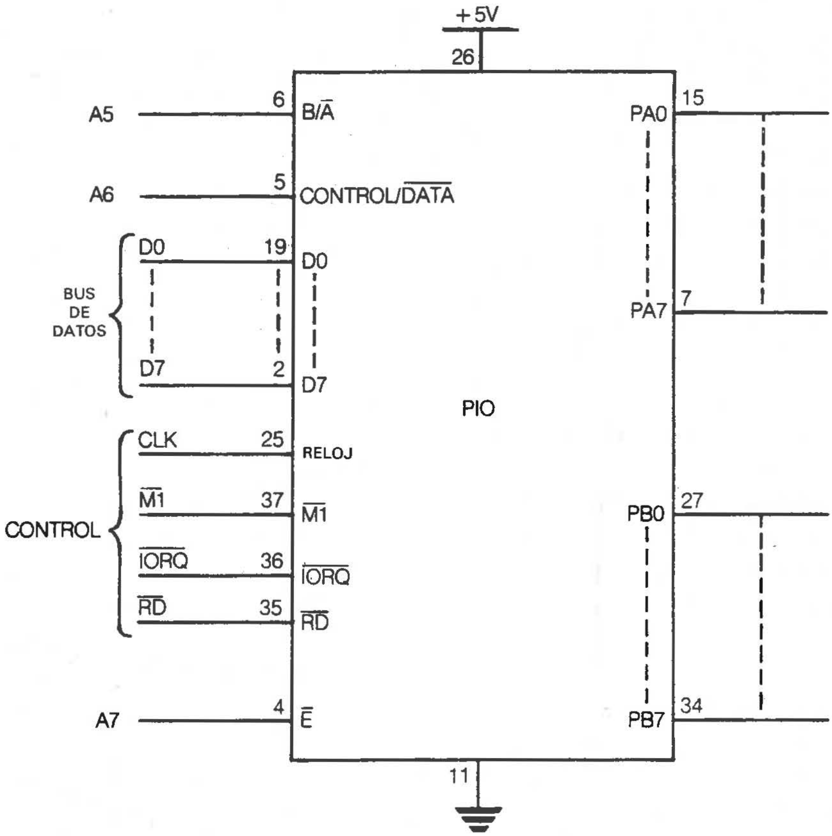
Interface del PIO

La Figura 3.2 muestra las conexiones básicas que se requieren para conectar el PIO al Spectrum. Se necesitan algunos componentes adicionales para controlar dispositivos y para tomar de ellos datos de entrada. Veremos, por ejemplo, cómo pueden añadirse relés para controlar dispositivos electrónicos generales, como se hizo en el Capítulo 2. Pero antes deberán añadirse al PIO algunas lamparitas y pulsadores, de manera que podamos experimentar las causas y efectos sobre las luces de la manipulación de dichos pulsadores, antes de escribir programas para el PIO.

En la Figura 3.3 aparecen estos componentes extras. Se han elegido cuatro líneas del PIO, dos de las cuales serán configuradas como salidas y las otras dos como entradas. Las líneas PAØ y PA1 (terminales 15 y 14) serán entradas, y por ello conectadas a los pulsadores para permitir cambiar sus estados lógicos manualmente. Las líneas PA2 y PA3 (terminales 13 y 12) son salidas y se conectan a los transistores que hacen que se enciendan o apaguen los leds.

La Figura 3.4 muestra el esquema de montaje del circuito, pero sin el detalle del cableado. Para el montaje deben seguirse los mismos principios generales que se establecieron en el Capítulo 2 para las placas de pistas impresas. Se deben utilizar siempre que sea posible zócalos para los circuitos integrados, que para el PIO es del tipo de 20 + 20 terminales (40 terminales "dual in line"). El mejor modo de efectuar el cableado en este tipo de circuito es el "wrapinado". Se trata de un cableado con hilo muy sólido y delgado que no sobresale demasiado de la placa. Para efectuarlo hay que tener un poco de práctica en el enrollado de los terminales, pero es muy conveniente aprender a utilizarlo. No se deben dejar longitudes excesivas en los extremos de los hilos. Debe soldarse el hilo limpiamente y cortar sin dejar los extremos sueltos, sino cortando el sobrante. Cuanto más meticulosos seamos en la ejecución de estos detalles, más garantías tendremos de evitar fallos. No se debe utilizar una longitud grande de cinta de conexión, pues las líneas de bus no están preparadas para enviar las señales eléctricas a gran distancia. Recomendamos una longitud de unos diez o quince centímetros.

Una vez terminado y verificado totalmente se enchufará al Spectrum. Sin insertar aún el PIO en su base, inspeccionaremos el extremo del circuito impreso del Spectrum que hace las funciones de base para el conector. Este posee contactos flexibles que aprisionan los extremos de las pistas de cobre. Cuidaremos de colocar el conector exactamente en su lugar. Para ello nos guiaremos por la ranura,



A6	A5	REGISTRO
0	0	REGISTRO DE DATOS A
0	1	REGISTRO DE DATOS B
1	0	REGISTRO DE CONTROL A
1	1	REGISTRO DE CONTROL B

Fig. 3.2. Interconexión del PIO.

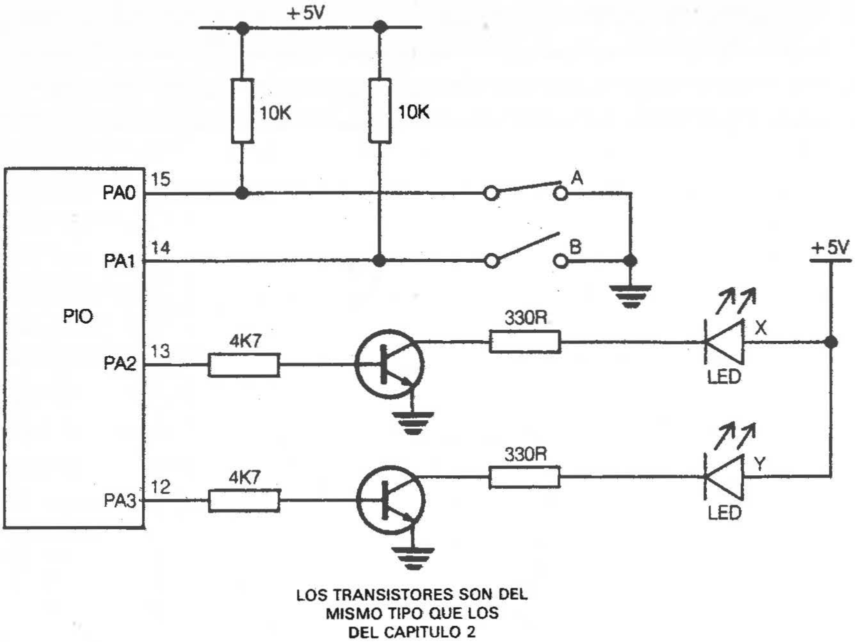


Fig. 3.3. Prácticas con el PIO.

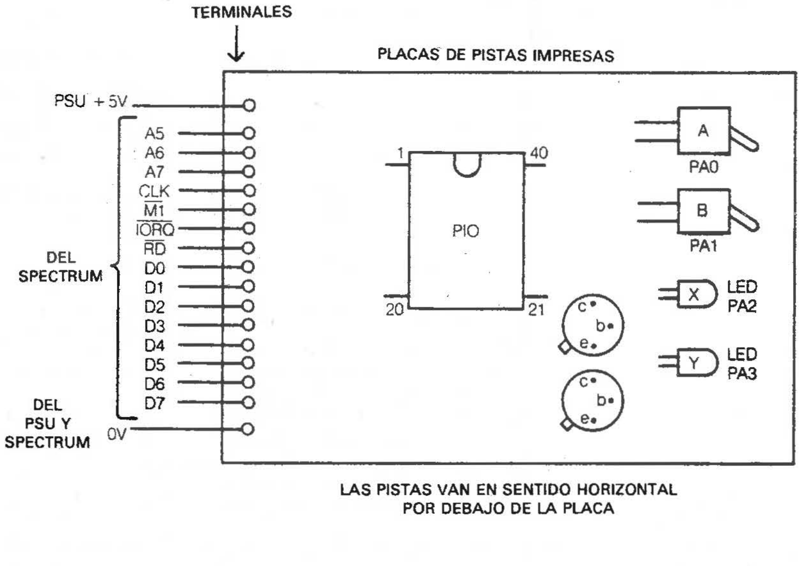


Fig. 3.4. Montaje del interface para el PIO.

que sólo puede estar en una posición. Cuidaremos, además, de que los contactos del conector coincidan lo mejor posible con los correspondientes extremos de las pistas del circuito impreso, para que no se produzcan cortocircuitos entre pistas adyacentes y asegurar buenos contactos.

Cuando estemos seguros de que lo hemos colocado correctamente, situaremos la placa en una superficie aislante limpia, para evitar cortocircuitos. A continuación, y con el PIO todavía fuera de la placa, encenderemos el Spectrum. Si no apareciera el conocido mensaje inicial en la pantalla, se debe desconectar inmediatamente y verificar todo lo que hemos hecho; probablemente existirá un cortocircuito que habrá de descubrirse antes de volver a intentarlo. Una vez esté todo bien, con el Spectrum apagado, insertaremos el PIO, encenderemos el Spectrum y, si el mensaje inicial aparece, podemos continuar.

Utilización del PIO

Para poner en funcionamiento el lado A del PIO, sólo es necesario enviar una secuencia particular de bytes de control para acceder a la dirección 95, correspondiente al registro de control para el lado A. De la misma forma se direccionará el 127 para el lado B. Aunque la operación del PIO puede ser muy compleja, nos conformaremos para nuestras aplicaciones con la utilización de dos puertos I/O solamente.

Para hacer funcionar uno de los lados del PIO, el primer paso será escribir el byte "255" y así poder controlar los registros del PIO. Esto es una especie de llave mágica. Conociendo este "ritual" podremos llegar a leer datos en el chip. Los bits del siguiente byte que se escribe en el registro de control, los utilizará el PIO para determinar qué líneas son entradas y cuáles salidas. Cuando hay un uno, el bit correspondiente de I/O es de entrada. Un cero significa salida.

Por ejemplo, si queremos que todas las líneas PA sean salidas, se deberá teclear lo siguiente:

```
OUT 95,255
OUT 95,0
```

Con ello lo que se hace es escribir el 255 y posteriormente el cero en el registro de control del lado A. Como el cero decimal corresponde en binario de 8 bits al ØØØØØØØØ, todas las líneas PA estarán

configuradas como salidas. Si hubiésemos escrito en la segunda sentencia, por ejemplo, 239, que en binario es 111Ø1111, haríamos que todas las líneas PA fuesen entradas, excepto la cuarta.

El próximo paso podría ser enviar otro byte al registro de datos del lado A. Estos bits en el byte que corresponde a líneas de salida aparecerán en las líneas correspondientes, no afectando a las de entrada. El programa siguiente hará que todas las líneas del lado A sean salidas y, a continuación, todas se activarán:

```
10 OUT 95,255
20 OUT 95,0
30 OUT 31,255
```

Como, a consecuencia de la ejecución de este programa, se activan todas las líneas del lado A, deberá cuidarse que los pulsadores no estén cerrados, pues en ellos habría nivel lógico alto y sería derivada a tierra la tensión proporcionada por el PIO, que podría dañarse.

Cambiando el número 255 de la instrucción 30, que comanda el puerto 31, podemos cambiar el estado de las lamparitas, encendiéndolas y apagándolas a voluntad. Se puede también conocer el último número enviado a las salidas del PIO. Para ello, si tecleamos

```
PRINT IN 31
```

veremos que aparece el número 255 en la pantalla una vez ejecutado el programa anterior. Conviene hacer esto mismo cambiando cada vez la línea 3Ø.

Una sentencia IN es la que se utiliza para leer los estados de las líneas de entrada. Tan sólo aquellos bits del registro de datos que estén definidos como entradas se leerán como datos de entrada: los demás serán leídos como el último estado que hayan tenido. Ahora veremos cómo podemos leer los estados de los pulsadores de la Figura 3.3.

Hemos llamado A y B a los pulsadores, y X e Y a las lamparitas (LEDs).

El primer paso será inicializar el PIO, enviando 255 por el puerto del registro de control A (OUT hacia la dirección 95), seguido por un byte para seleccionar las líneas que deseamos sean entradas o salidas. Necesitamos dos líneas de entrada para los pulsadores, y dos líneas de salida para las lamparitas. Así pues, el siguiente número (el que sigue al 255) enviado hacia el registro de control, deberá tener un valor binario de la forma

xxxx∅∅11

donde x puede tener cualquier valor, ya que las líneas PA correspondientes a esos bits no están conectadas, y por ello es irrelevante. Cualquier número cuyos últimos cuatro bits sean iguales al anterior nos valdrá; por ejemplo, el ∅∅∅∅∅∅11, que corresponde con el 3 decimal.

Después de enviar los dos bytes de control, el estado de las luces y pulsadores pueden ser alterados y leídos, respectivamente, en la dirección 31, y como los pulsadores están conectados ahora a líneas de entrada, no existe riesgo de cortocircuito si actuamos sobre ellos como queramos.

Veamos primero las lamparitas. Para encender la luz X solamente debemos enviar un 1 por la línea correspondiente al bit número 2, pero no a la línea del bit 3. Esto puede conseguirse con cualquier número de la forma

XXXX∅1XX

siendo x, como antes, cualquier cosa. Por ejemplo, nos servirá el número ∅∅∅∅∅1∅∅, es decir, el 4 decimal. Comprobemos ahora que el 8 decimal encenderá el otro LED, y que el 12 encenderá los dos. Lo importante de esta experiencia es que nuestra manera de comunicar órdenes de control es por medio de números decimales introducidos por medio de sentencias BASIC.

Para leer el estado de los pulsadores aplicaremos el mismo principio. Intentemos ejecutar el siguiente programa, observando los números que aparecen en la pantalla al cambiar el estado de los pulsadores:

```
10 OUT 95,255
20 OUT 95,3
30 OUT 31,0
40 CLS
50 FOR I=1 TO 100
60 PRINT IN 31;
70 NEXT I
80 GO TO 40
```

Con la línea 3∅ sencillamente ponemos todas las salidas a cero para asegurar que el número que aparezca es un número definido por la nueva situación.

Cuando se deban utilizar conjuntamente entradas y salidas, no será posible poner las salidas en un solo estado como hemos hecho

antes. Además, si se utilizan conjuntamente, habrá que hacer ciertos análisis para extraer los bits de entrada del resto y poder determinar así los estados de tales bits de entrada.

Como podemos observar, cuando se lee un puerto del PIO, el número que aparece está compuesto por dos tipos primarios de datos. Los bits correspondientes a las entradas indican el estado de esas entradas y los de salida indicarán el estado de las salidas mientras dure el efecto producido por la última sentencia OUT enviada a ese puerto. Supongamos que el pulsador A está cerrado, como se muestra, y el B está abierto. Supontamos también que ambas lamparitas están encendidas. El equivalente binario del número decimal que debe aparecer tras la ejecución de la sentencia.

```
PRINT IN 31
```

será

xxxx11Ø1

A causa de la indeterminación de los 4 primeros bits, el número decimal que aparecerá podrá ser cualquiera de los 16 números que terminan en 11Ø1. Esto impide conocer directamente el estado de las líneas a partir de dicho número decimal. Sería mucho mejor que apareciese directamente en binario, pero desgraciadamente en BASIC no se utiliza, y tendremos que realizar nosotros la conversación (ver Apéndice 1) utilizando divisiones sucesivas.

Como ejemplo, supongamos que la lectura en el puerto 31 nos proporciona el número 29; tendremos que realizar las siguientes divisiones:

29/2 = 14 resto: 1 = bit Ø (estado#del#pulsador#A)
14/2 = 7 resto: Ø = bit 1 (estado#del#pulsador#B)

Con este procedimiento es suficiente y puede incluirse dentro de nuestro programa de control BASIC para leer inmediatamente el estado de los pulsadores. El siguiente programa verifica el estado de los pulsadores continuamente y transfiere esos estados a los LEDs:

```
10 REM *** INICIALIZACION DEL PIO
20 OUT 95,255
30 OUT 95,3
```

```
 40 REM *** LECTURA DELOS ESTADOS DE
    PULSADORES
 50 LET X=IN 31
 60 REM *** CALCULO DE BITS
 70 LET Y=INT (X/2)
 80 LET BIT0=X-2*Y
 90 LET BIT1=Y-2*INT (Y/2)
100 REM *** SI BIT0=0 ENTONCES
110 REM *** PULSADOR A ABIERTO,ETC
120 REM *** AHORA PONER LUCES
130 LET Z=4*BIT0+8*BIT1
140 OUT 31,Z
150 REM *** REPETICION
160 GO TO 50
```

Mientras este programa se mantenga en ejecución, podrán cambiarse los pulsadores y observar su efecto sobre las luces. Las líneas 7∅, 8∅ y 9∅ muestran cómo se tratan los números decimales para deducir los binarios que nos interesan. La línea 13∅ nos da una idea de cómo pueden convertirse los unos y ceros en estados de las lamparitas; por ejemplo, si el bit 1 es un 1, entonces la línea 13∅ añadirá el producto 1*8 al valor de la variable Z, y ésto dará como resultado que aparezca un 1 en el bit 3 para encender el LED Y.

Con estos principios, ya podemos escribir nuestros propios programas y experimentar con este sencillo circuito, lo que nos proporcionará una experiencia valiosísima para el control de líneas I/O en general.

El ratón robot

Durante muchos años, el ratón robot ha sido un símbolo del robot móvil más sencillo. Una forma muy sencilla de realizarlo es por medio de un chasis provisto de dos ruedas motrices en la parte de atrás y una sola rueda directriz delante. En esta sección se describe cómo puede realizarse el control de un dispositivo de este tipo utilizando sencillos circuitos electrónicos. Se trata de un buen experimento de aplicación robótica, pues comprende no sólo el control de un dispositivo mecánico propiamente dicho, sino también la utilización de realimentación desde el mismo, que es lo que hace que el control sea "inteligente". Utilizando el ratón se puede practicar con procesos sencillos como luego sugeriremos, siendo el primer paso para el aprendizaje de los métodos que le permitirán el control de otros robots más complicados.

La Figura 3.5 muestra una posible disposición mecánica. La marcha hacia adelante y atrás se consigue por medio de un motor de corriente continua que mueve el eje motriz a través de un sencillo engranaje de ruedas perpendiculares (piñón-corona). Se puede usar, desde luego, cualquier mecanismo que se encuentre en tiendas de modelismo. La dirección de la rueda delantera se consigue por medio de otro motor de corriente continua que mueve un eje vertical al que está solidaria la rueda, a través de un sistema de tornillo sin fin. En el eje de dirección se montarán unas palanquitas que accio-

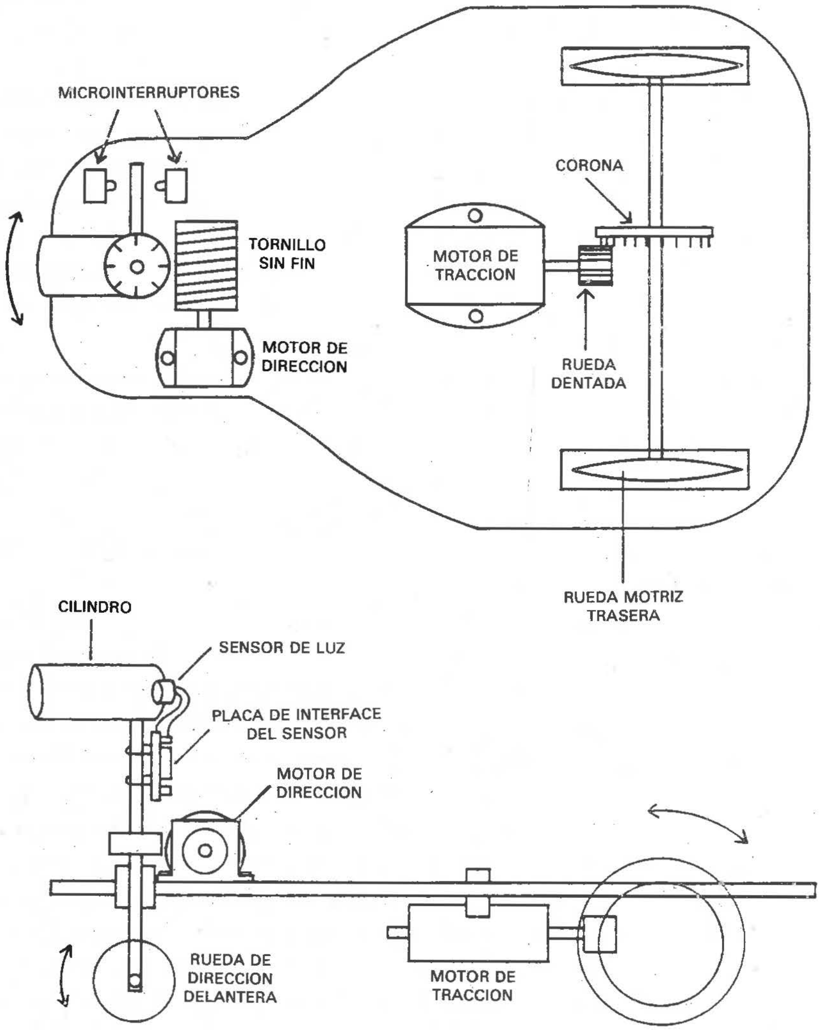


Fig. 3.5. Ratón robot

nen unos microinterruptores "fin de carrera", cuando el giro del eje lleje a su fin, tanto en un sentido como en otro. El tipo de motor que se instale decidirá la potencia que se necesita para accionarlos. Lo ideal sería poder utilizar baterías recargables que irían montadas en el propio ratón robot.

Habrá que realizar el cableado adecuado para conectar el robot con el Spectrum, puesto que hay que transportar las señales de control y de realimentación de uno a otro. En una realización más sofisticada se emplearía para ello un sistema de transmisores y receptores de infrarrojos, lo que liberaría al robot de mantener una conexión física con el ordenador.

La Figura 3.6 muestra un circuito para controlar la dirección y la tracción por medio de relés. Los relés que controlan la columna de

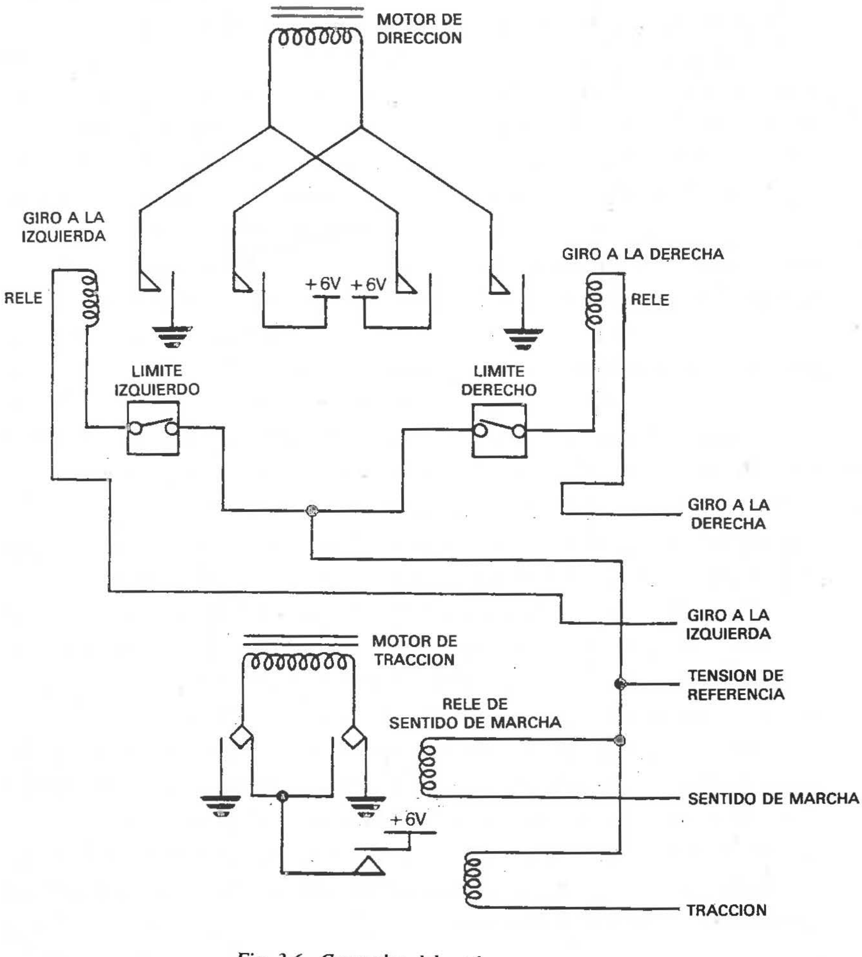


Fig. 3.6. Controles del ratón

dirección tienen dos contactos cada uno, y el relé de marcha del tipo de doble polaridad. Todos los contactos de los relés están dibujados en el estado de reposo. Debemos inspeccionar cuidadosamente este diagrama antes de continuar e imaginar los efectosa que ocasionarían las distintas aperturas y cierres de los contactos de los relés dibujados.

El motor de la dirección permanecerá parado hasta que uno de los dos relés de dirección se actúe. Cada uno de estos relés lo que hace es poner ØV y 6V (si es ésta la tensión de las baterías recargables) en las conexiones del motor de dirección para que éste gire en uno u otro sentido. Es importante, pues, que en cualquier momento esté accionado un sólo relé o ninguno, pues de estar ambos accionados, se produciría un cortocircuito de la batería de alimentación. Por ello, y en previsión de errores, es conveniente montar un fusible adecuado en la línea de alimentación de las baterías.

Si uno de los relés de dirección permanece continuamente activado, el motor funcionará haciendo que la columna de dirección gire en un sentido, permanentemente, hasta alcanzar al "fin de carrera", que abrirá el circuito del motor, con lo que finalizará el giro.

El sentido de la marcha y la alimentación para el motor de tracción se controlan por medio de otros dos relés. Los cuatro relés se conectan a un voltaje común de mando, lo que significa que hay que enviar al ordenador 5 hilos de control, puesto que la alimentación de los motores se realiza por medio de baterías incorporadas en el robot, si encontramos la manera de hacerlo.

Pero hemos dicho anteriormente que conviene realizar el control de manera inteligente, para lo cual habrá de disponer de la consiguiente realimentación que informe al ordenador del estado del robot. Podríamos hacerlo según distintas filosofías, por ejemplo, dotándolo de detectores de "choque" delante y detrás, de modo que se pueda invertir la marcha si se detecta un obstáculo. Esto se podría conseguir por medio de microinterruptores similares a los "fin de carrera" utilizados en la dirección del robot. Pero también se podrían intentar colocar detectores de calor, de luz, de sonido, etc., según la aplicación a la que fuera destinado el robot.

Hemos elegido para nuestro experimento un detector luminoso que proporcionará una indicación positiva cuando la luz que llegue a él se encuentre por debajo de un cierto umbral de intensidad. Existen detectores de intensidad luminosa de este tipo en el mercado; no obstante, a continuación describimos un detector que podemos construirnos nosotros mismos.

Detección de luz

El mejor lugar para colocar un detector luminoso creemos que puede ser la parte superior de la columna de dirección, como está dibujado en la figura 3.5. Cuando la dirección gira, lo hace también el detector, por lo que estará siempre orientado en la dirección a la que se dirige el robot, en cualquier momento. En la figura 3.7 podemos ver la fotocélula montada en el interior de un cilindro, para asegurar que la luz que llega a la fotocélula es la que procede del lugar al que se dirige el robot. Podría añadirse un sencillo sistema de lentes para concentrar la luz más precisamente a incrementar así la sensibilidad y directividad del dispositivo detector de luz.

Existen muchos tipos de células que pueden proporcionar una indicación electrónica del nivel luminoso que reciben, descritas en libros de electrónica general, que nos pueden servir para nuestros propósitos. Aquí mencionaremos tan sólo dos de esos tipos. La LDR (resistencia dependiente de la luz) es uno de los tipos más antiguos de detectores luminosos. Su funcionamiento consiste, simplemente, en que el valor de la resistencia disminuye cuando la intensidad de luz aumenta. La Figura 3.7 nos muestra un circuito para la LDR normal, conocida como CDS (célula de sulfuro de cadmio), colocada en una disposición típica, en placa Veroboard. La LDR más conocida correspondiente a este tipo es la ORP12, que se encuentra fácilmente en el mercado. Aquí la vemos conectada a un circuito que utiliza un tipo especial de circuito integrado llamado Op-amp (amplificador operacional). El más conocido y fácilmente encontrado en el comercio de este tipo de circuito integrado es el 741, que puede conseguirse en la misma versión que aparece en la figura, en encapsulado de 4 + 4 terminales en línea.

La resistencia variable plana, miniatura dibujada en la figura, tiene tres terminales. Uno de ellos se denomina "cursor" y en este circuito está conectado a uno de sus otros terminales, y a Ø V. Variando esta resistencia varía el umbral de detección de la luz. La salida del circuito variará de Ø a 1 según el nivel de luz exceda, o no llegue a la banda de umbral, asegurándonos que no existen en la salida voltajes intermedios que puedan confundir a la línea de entrada del ordenador. Para conseguirlo, el 741 se dispone en una configuración llamada *disparador de Schmitt*. Debe experimentarse con distintos valores de la resistencia para obtener el mejor resultado para las condiciones de iluminación en las que se va a utilizar el ratón robot.

El otro tipo de interruptor luminoso está basado en el uso de un

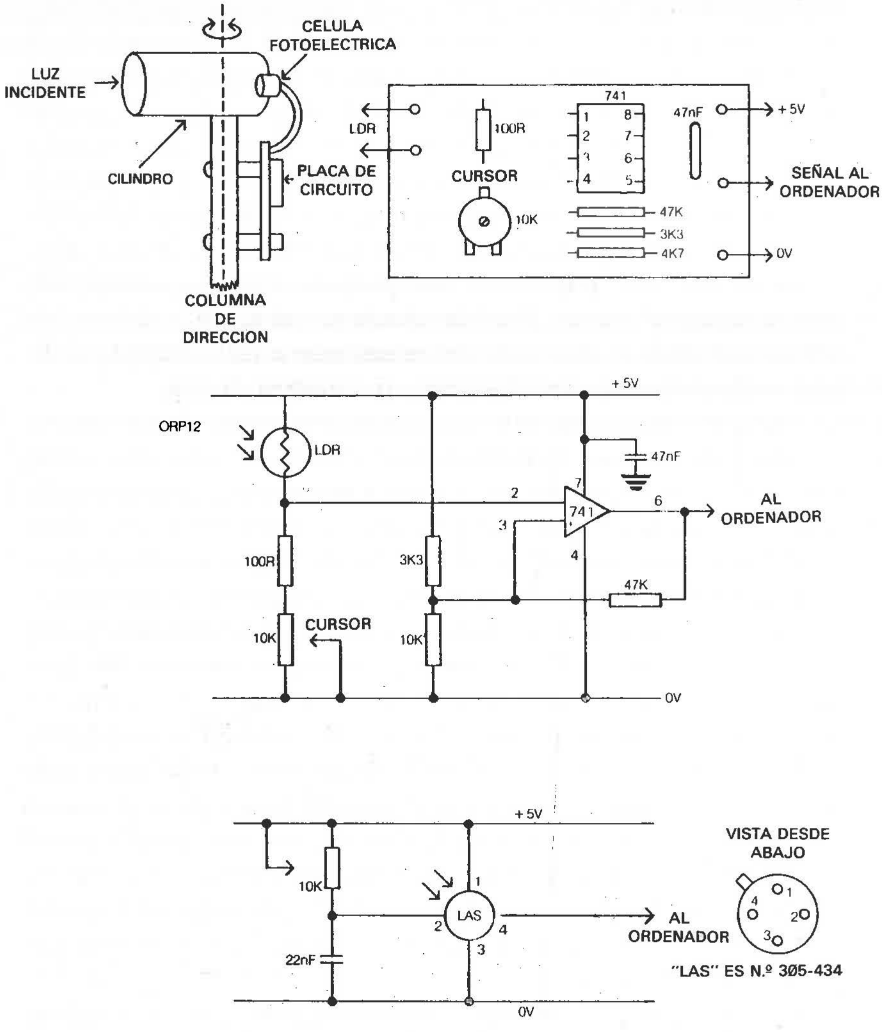


Fig. 3.7. Interruptor luminoso

dispositivo que se comercializa encapsulado en la misma forma que los transistores, llamados LAS (interruptor activado por la luz). Este dispositivo proporciona a su salida un 1 o un Ø, dependiendo del nivel de la luz y del umbral establecido por el potenciómetro de 10 K.

Tanto si utilizamos uno u otro circuito, deberá montarse en un trozo de placa Veroboard, inmediatamente debajo de la propia célula, aislando la superficie inferior de la placa para evitar cortocircuitos.

La señal de esta placa y las líneas de alimentación para ella, deberán llegar a través de cables finos, para no interferir con el movimiento del motor de dirección. Dichos hilos se conectarán a la placa del PIO.

La Figura 3.8 muestra la interconexión (o interface) del ratón robot con el Spectrum, utilizando el PIO. Los relés mostrados deberán trabajar con 5V, siendo la resistencia de sus bobinas del orden de 1ØØ ohmios o mayores. Para desacoplar el ruido electrónico que pudieran producir se conectan condensadores de 1ØØ nF entre cada uno, y un diodo de protección como se explicó en el capítulo anterior, si los relés no son del tipo que llevan el diodo incluído. Los transistores son como los del capítulo anterior. En la Figura 3.6 existe una línea de voltaje común para relés. Es la línea de +5V que aparece en la Figura 3.8 y llega de la tarjeta de alimentación del PIO que está construída tal como se explicó en el Capítulo 2.

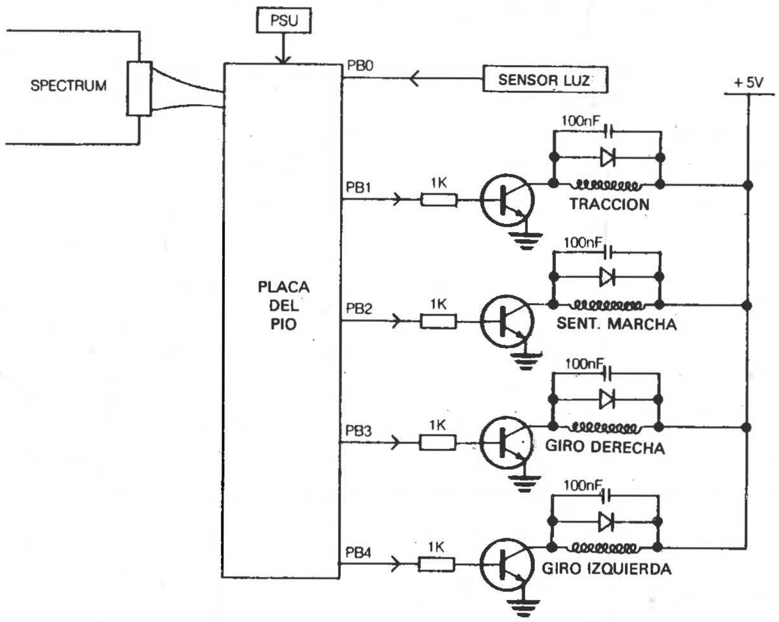


Fig. 3.8. Interconexión del ratón con el Spectrum

Por contraste, en esta aplicación se utiliza el lado B del PIO. El sensor luminoso se conectará a PBØ, para poder analizar más fácilmente su estado a partir de los datos presentados en la pantalla. Los relés se activan por los estados altos PB1-PB4.

Operación con el ratón robot

El ratón robot aquí descrito es capaz de realizar muchos tipos de funciones. Una vez que nos hayamos familiarizado con el sistema básico, podremos realizar ampliaciones. De momento, vamos a seguir utilizando las mismas entradas y salidas que ya conocemos, para gobernar el robot de manera que se oriente hacia una fuente luminosa.

La primera tarea a realizar es ajustar el umbral de luz de tal manera que el ratón robot pueda "verla" y distinga entre su visión directa y la media luz de su entorno. Lo haremos mediante la ejecución del siguiente programa:

```
 10 REM ** ACTIVACION DEL LADO B DEL PIO
 20 OUT 127,255
 30 REM ** PB0 ENTRADA, RESTO SALIDA
 40  OUT 127,1
 50 REM ** TODAS LAS SALIDAS A 0
 60 OUT 63,0
 70 REM ** ENTRADA Y PRESENTACION
 80 CLS
 90 FOR I=1 TO 100
100 PRINT IN 63;
110 NEXT I
120 GO TO 80
```

Conforme variemos el potenciómetro, encontraremos una posición para la cual, al ensombrecerse la luz, cambiará el número. Si así ocurre, es que funciona correctamente. De este modo podremos usar la realimentación para controlar el motor de dirección.

Para conmutar los relés, han de enviarse números hacia el puerto 63. El relé para la tracción está conectado a PB1, que corresponde al bit 1 del bus de datos. La instrucción

OUT 63,2

activará el relé. La siguiente tabla muestra los números que deben ser enviados hacia el puerto 63 según las funciones que queramos realizar.

Número de salida	*Acción*
2	Activa relé de tracción
4	Cambio de sentido
8	Giro a la derecha
16	Giro a la izquierda

Cualquiera de ellos pueden enviarse juntos, excepto la combinación de giro a derecha e izquierda simultáneos, pues se fundiría el fusible de alimentación de los motores. Por ejemplo, al enviar el número 14 = 2 + 4 + 8, el ratón robot deberá marchar hacia delante y simultáneamente girar a la derecha. Para ello hay que teclear:

```
OUT 63,14
```

Esto es todo lo que hay que hacer para conectar motores, es decir, hay que decidir qué relé gobierna el giro a la derecha, cuál a la izquierda, el sentido de la marcha, etc. En general, se debe procurar que las conexiones se hagan de tal modo que la situación de reposo de los relés se correspondan con los estados más frecuentes. Por ejemplo, el relé de tracción en situación de reposo hará que el ratón robot se mueva hacia adelante. Con esto ahorramos energía y utilización del relé.

Debemos practicar activando y desactivando funciones, para alcanzar alguna experiencia con el método de control antes de continuar. Nótese que si la dirección llega al final en cualquiera de los dos sentidos de giro, no podrá seguir siendo dirigido más allá, pero sí se le puede hacer que gire en el otro sentido, con lo que el microinterruptor volverá a cerrar el circuito de nuevo.

Puesto que no hay realimentación que nos indique el estado de los microinterruptores "fin de carrera", en esta versión básica, deberemos utilizar algún procedimiento, como temporizaciones, para limitar el tiempo de gobierno de un giro. Para llevarlo a cabo, se deberá orientar la columna de dirección hacia una fuente luminosa y girar hacia atrás y adelante hasta detectar un nivel de luz suficiente. Se llevará la columna de dirección en un sentido hasta el final de su recorrido, dejándola que gire en ese sentido el tiempo necesario. En el programa se ejecuta un lazo, durante el cual estará siendo leído PBØ continuamente hasta que se detecte la luz. Mediante varias pruebas determinaremos cuántas veces se debe ejecutar el bucle para

asegurar que la columna ha llegado al fin, que dependerá del dispositivo mecánico empleado. Una vez que llegue al fin, se comienza a girar en sentido contrario, sin dejar de monitorizar PBØ. Si no se detecta la luz, el lazo continuará sin verificarse. Cuando la luz pueda aparecer, debe pararse el giro de la dirección y seleccionarse la marcha hacia delante. Esto hará que automáticamente se coloque la dirección hacia la luz. El ratón robot se moverá alrededor de un arco, mientras el detector luminoso esté lejos de la luz. El motor de marcha entonces debe pararse, y repetirse el proceso de la detección y dirección. En el siguiente apartado se describe un programa que hace que esto se realice.

Desde luego, esta función puede llevarse a cabo de diferentes maneras, y después de practicar con el robot ratón podremos perfeccionar la rutina. Por ejemplo, cuando la luz se encuentra, la tracción podría continuar y la dirección podría moverse de un lado a otro para mantener la luz en su trayectoria. De esta manera, el ratón robot estaría la mayor parte del tiempo en la línea de la luz.

Programación del ratón robot

La Figura 3.9 muestra un diagrama de la actividad ya descrita, en forma de *carta de flujo*. En esta carta aparece el programa de forma visual facilitando su comprensión y descripción. Se han utilizado tres tipos de símbolos o bloques. El bloque rectangular indica que su contenido es una instrucción. El rombo contiene una decisión, y tiene dos salidas llamadas "si" y "no". El primer bloque situado en la parte superior es rectangular con los lados redondeados, y significa el comienzo, la primera instrucción de la rutina. Mediante esta carta podemos seguir el programa con facilidad, como veremos ahora.

El primer paso es activar el PIO indicando las líneas correctas de entrada y salida. A continuación hacemos cero la variable N que va a ser empleada como contador de los lazos. La siguiente instrucción simplemente asegura que el motor de tracción está parado. El ratón robot está a la espera de ver la luz y dirigirse hacia ella. El siguiente bloque es una decisión: si no se detecta luz, el contador avanza, y como el valor de N no ha alcanzado aún el 2ØØ, la luz continuará siendo monitorizada, moviéndose la columna de dirección hacia la derecha. Esta situación continúa hasta que la luz sea detectada o el contador supere el valor superior, que hace que la columna de dirección sea girada ahora hacia la izquierda. El valor de 2ØØ como tope del contador se ha establecido arbitrariamente, por lo que debere-

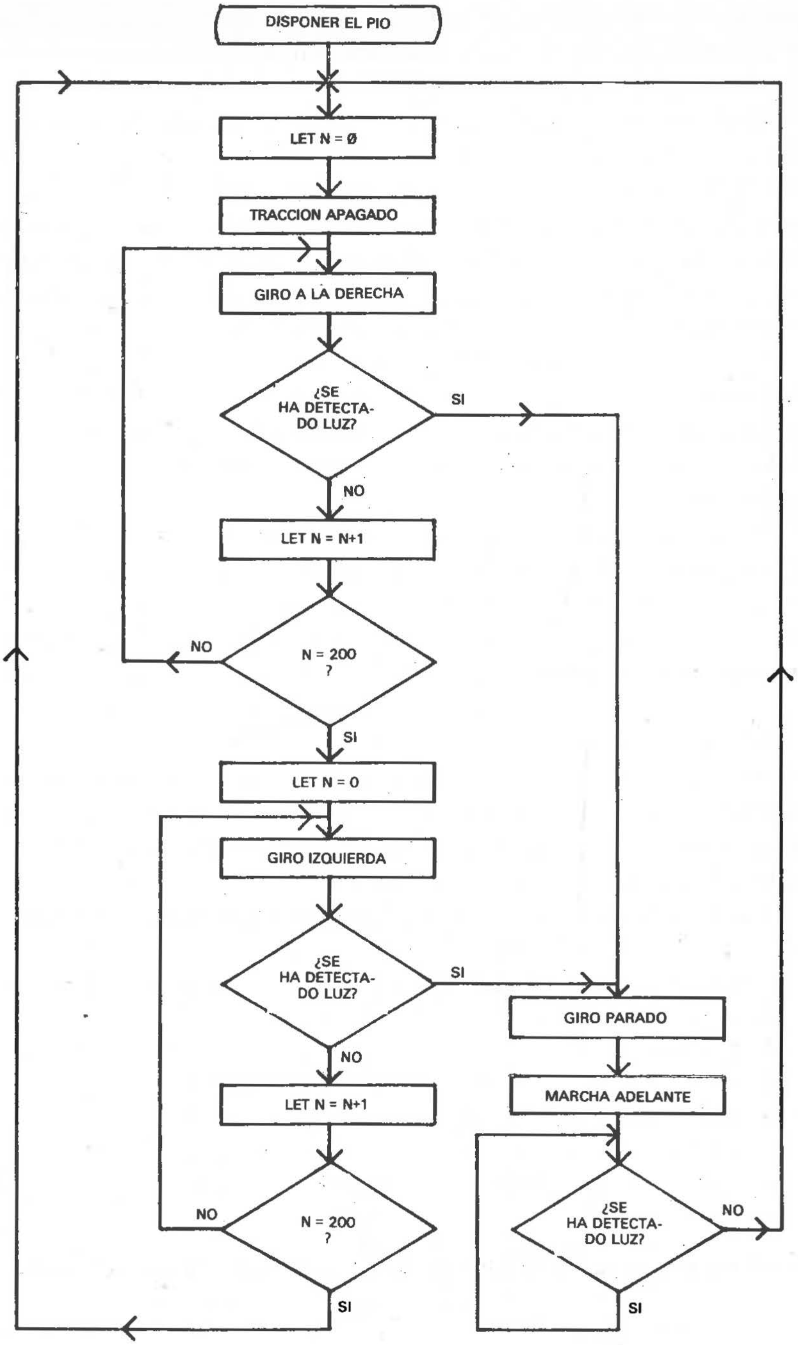


Fig. 3.9. Carta de flujo para el control del ratón

mos poner el que deduzcamos de las pruebas, para asegurarnos que la dirección alcanza los topes a ambos lados en el tiempo que tarda el contador en llegar a ese valor.

Si el contador supera este valor cuando va hacia la izquierda y se está verificando el detector de luz, utilizaremos de nuevo un contador para limitar el tiempo para el cual esta función debe llevarse a cabo. Si no se detecta luz en esta dirección antes de que el contador llegue a 2ØØ, la rutina comenzará de nuevo, reponiéndose el contador y comenzando el giro a la derecha. Así permanecerá funcionando continuamente hasta que se detecte la luz.

Si la luz se detecta antes de que el contador llegue al fin, el motor de dirección se parará, y el de tracción se moverá hacia adelante. La detección de luz todavía se está verificando, por lo cual, si desapareciera, la rutina comenzaría desde el principio otra vez, con el motor de tracción parado.

La detección de luz puede verificarse dentro de la rutina leyendo el estado de la línea PBØ; esto es, leyendo el número que aparece procedente del puerto 63. Este número decimal puede ser par o impar. Si es par, querrá decir que el bit es el Ø y, por tanto, PBØ es cero; si es impar, será un uno. Si se ha elegido un circuito con LDR, entonces PBØ a nivel Ø significará que la luz ha sido detectada y viceversa.

El programa continúa sin más control hasta que hagamos un BREAK, lo que significa que si el robot detecta luz, irá hacia ella para siempre y no parará cuando llegue a ella. Habrá que tener ésto en cuenta cuando utilicemos esta rutina.

De la carta de flujo anterior se deduce el programa siguiente:

```
 10 REM ** ACTIVAR LADO B DEL PIO
 20 OUT 127,255
 30 OUT 127,1
 40 OUT 63,0
 50 REM ** CONTADOR
 60 LET N=0
 70 REM ** DERECHA Y PARADO
 80 OUT 63,8
 90 REM ** VERIFICAR LUZ
100 LET K=IN 63
110 IF 2*INT (K/2)=K THEN  GO TO 500
120 LET N=N+1
130 REM ** REPETIR LAZO HASTA ALCANZAR LIMITE
140  IF N<200 THEN  GO TO 60
150 REM ** IZQUIERDA Y REPETIR COMO ANTES
```

```
160 LET N=0
170 OUT 63,16
180 LET K=IN 63
190 IF 2*INT (K/2)=K THEN  GO TO 500
200 LET N=N+1
210 IF N<200 THEN  GO TO 160
500 REM ** PARAR DIRECCION Y AVANCE
510 OUT 63,2
520 REM ** CONTINUAR SI LA LUZ CONTINUA
530 LET K=IN 63
540 IF 2*INT (K/2)=K THEN  GO TO 530
550 GO TO 60
```

El mejor método para estudiar este programa y utilizarlo como base de nuestros propios experimentos es simplemente utilizarlo tal como está escrito, palabra por palabra. Por ejemplo, se nos podría ocurrir mejorarlo añadiendo un contador, utilizando pocas líneas al final del mismo con objeto de limitar el tiempo que el robot sigue la luz y, por consiguiente, evitar que se acerque estúpidamente a ella.

Otro refinamiento que se nos podría ocurrir sería un reloj de tiempo real, como el programa que viene en el manual del Spectrum, que haría que el ratón realizarse sus actividades de acuerdo con alguna estructura temporal. Se nos podría ocurrir utilizar los dos bits de la izquierda de las líneas PB para obtener realimentación a partir del estado de microinterruptores en la columna de dirección, para que el ordenador esté informado de cuándo se alcanza el límite mecánico y no tener que recurrir al control de bucle abierto como contador de tiempo, como se ha hecho anteriormente. No obstante, si decidimos incorporar algunos de estos procedimientos, recordemos que si empleamos las líneas PB1 y PB2 como entradas, debemos definirlas como tales y deslizar las salidas dos posiciones, para hacer el cálculo más fácil.

Cualquier proyecto robótico puede ser ampliado en la medida deseada. El que nos ocupa ahora no es más que una base de partida para familiarizarnos con este tipo de control. Como ejemplo de posibilidad de expansión podríamos hacer, puesto que el robot se mueve por el suelo, un almacenaje de los valores de N o de cualquier variable que se introduzca para conocer en todo momento cuál ha sido el camino recorrido por el mismo, y utilizarlo para que realice el mismo camino en sentido contrario, o para que aprenda una ruta desde el comienzo hasta el lugar donde ha detectado la luz, o donde ésta se encuentra habitualmente.

La mejor manera de extensión software, desde luego, sería la utilización del código máquina, por medio de uno de los excelentes programas ensambladores existentes en el mercado, y controlar los bits de una manera más directa. El proceso de ejecución del programa sería mucho más rápido, aunque no notaríamos gran cosa la diferencia en este proyecto, pues los mecanismos del ratón probablemente son muy lentos comparados con la velocidad del cálculo del ordenador.

Periféricos en general

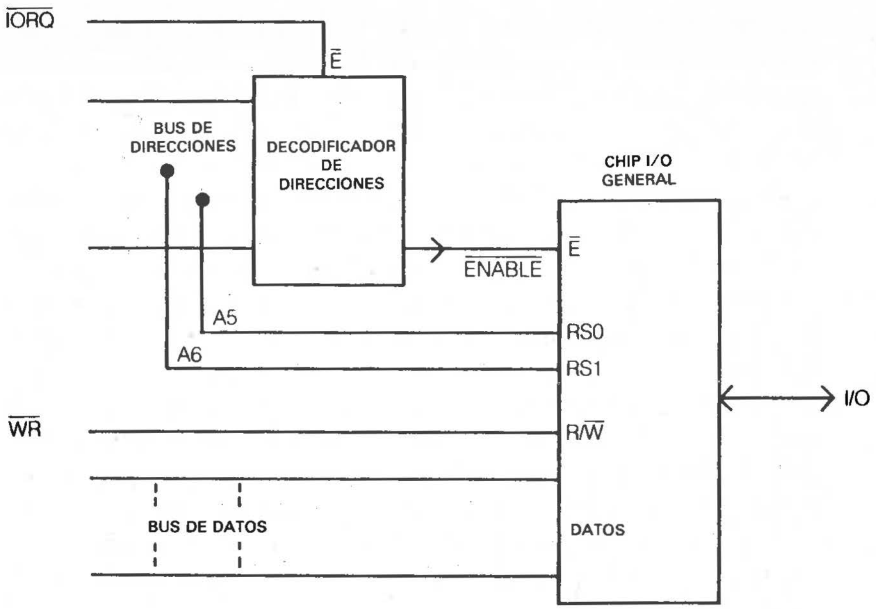
Hay algunos principios generales que podemos aplicar al diseño de nuestros propios periféricos, basados en el interface anterior para el PIO.

El primer principio es que la mayoría de los chips LSI (integrados a gran escala) tienen líneas I/O, un bus de datos, una línea Enable (habilitación), y/o algunos registros internos y una línea RD o WR o equivalente. En efecto, en general podemos hacerlo todo justamente con estos terminales. Algunos chips I/O, sin embargo, tienen algunas líneas además de éstas. Estas líneas extras se utilizan para aspectos más complejos del control del chip. Normalmente, las anteriores son suficientes para una aplicación simple.

La Figura 3.10 muestra un ejemplo de un chip I/O general. No tiene por qué ser un chip I/O. Puede haber sido diseñado para el interface con el ordenador de voltajes que varían analógicamente, o para líneas serie que envían los datos uno a uno. El propio I/O real no es importante como interface. Si el chip tiene registros internos realmente el carácter del I/O involucrado es más una materia de software, lo que debe ser escrito y leído en esos registros.

El diagrama muestra la forma de un periférico. Con esto y la explicación que sigue, ya deberíamos ser capaces de realizar el interface con la mayoría de los periféricos del Spectrum o de cualquier otro ordenador. El bloque que no ha sido explicado en detalle es el bloque de decodificación de direcciones. Simplemente lo que hace es reconocer direcciones enviadas al periférico y poner su línea de salida a nivel bajo para habilitar o activar el periférico. Veremos un ejemplo de esto más tarde. Los circuitos que hemos visto hasta ahora (la ULA y el PIO) han realizado esta función automáticamente, tomando las direcciones de las líneas correspondientes directamente, y decodificándolas sin necesidad de circuitos exteriores.

En la Figura 3.10 está dibujada la línea IORQ, que se utiliza para



	REGISTRO	A6	A5
	REGISTRO 3	1	1
	REGISTRO 2	1	0
	REGISTRO 1	0	1
DIRECCION DE BASE	REGISTRO 0	0	0

Fig. 3.10. Periférico general.

activar el decodificador de direcciones, para asegurar que solamente los comandos I/O van a activar el circuito. La dirección que aparece en el bus de direcciones o en algunas de las líneas del bus de direcciones, se utilizan para activar el propio chip I/O. Al mismo tiempo, el estado de la línea WR nos dirá cuándo el chip está considerando que es entrada o salida. Se ha dibujado WR conectado a un teórico terminal W/R, es decir un tipo normal de entrada lectura/escritura y, como muestra el diagrama, si existe un 1 ocurrirá la situación de lectura, y se es Ø, la de escritura. Esto nos recordará exactamente la línea WR del Z8Ø. Nótese que RD es redundante, en general, y muchos microprocesadores sólo disponen de una línea R/W, que es equivalente a la WR del Z8Ø.

Una vez se activa el chip, la transferencia de datos puede comen-

zar, pero es esencial que se alcance el registro interno correcto. El diagrama está hecho suponiendo que el número de registros internos es cuatro o menor, y que se direccionan utilizando los terminales RS1 y RS∅, que están conectados a A6 y A5. La tabla muestra cuántos estados diferentes en A6 y A5 proporcionan dirección a registros diferentes. La dirección más baja del grupo se denomina *Dirección de base* y es esta Dirección de base la que está determinada por la decodificación de direcciones. En otros sistemas se podrían usar A1 y A∅ como líneas de selección de los registros y de este modo obtener un mapa más claro. Esto no puede hacerse con el Spectrum.

Las demás conexiones que quedan son únicamente el bus de datos y las de las tensiones de alimentación. La fuente de alimentación proporciona normalmente + 5 V, siempre que el I/O al que se esté haciendo el interface no tenga sus propias tensiones especiales.

Esto da una idea general del interface I/O con el ordenador. La decodificación de direcciones se describe a continuación añadiendo alguna lógica electrónica que, además de utilizarse para la decodificación, sirva para expandir el proceso a más de un periférico a la vez. Por ejemplo, si necesitamos dos o más PIOs en un proyecto.

Decodificación de direcciones

Ya hemos visto que cuando deseemos activar bajo programa un dispositivo, ha de enviarse el número decimal apropiado correspondiente a un puerto de salida. Hemos visto en el Capítulo 2 cómo poniendo un bit en el puerto 254 se activa una línea de salida. La ULA hace esta función decodificando el bus de direcciones y de control de la CPU para asegurar que el puerto 254 ha sido activado en su momento. Ahora veremos cómo se hace esto, en general.

Por ejemplo, para poner el bit 3 del puerto 31 a 1, la CPU enviará el conjunto de bits:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit ∅
∅	∅	∅	∅	1	∅	∅	∅

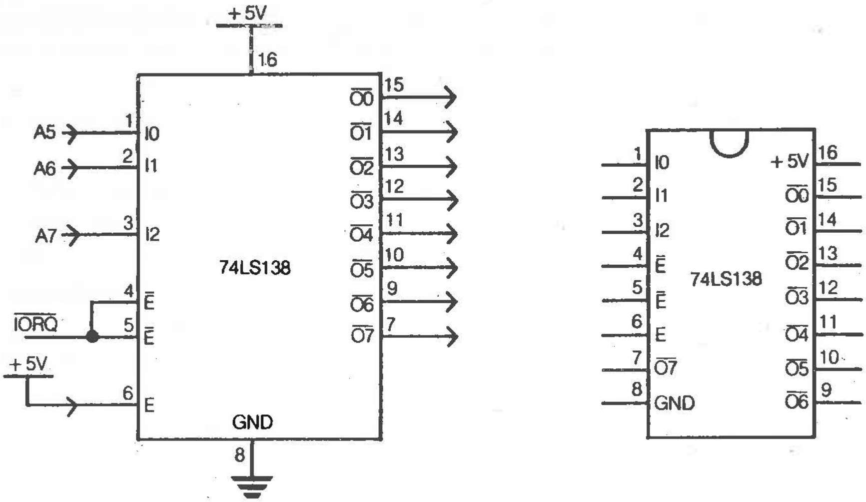
que es el ∅8 (hex). Este conjunto lo envía por el bus de datos. Para asegurar que la electrónica a la que se destina el puerto 31 se activa para recibir el byte, la CPU pone el conjunto de 8 bits más bajo del bus de direcciones, de tal forma que contenga el conjunto de bits correspondiente a 31 (dec), es decir ∅∅∅11111 (bin). Al mismo tiempo

que lo hace, debe reconocer que esto no es una transacción de memoria, para lo cual el IORQ debe ponerse a nivel bajo. Del mismo modo, RD debe ser alto y WR bajo, para mostrar que la CPU está escribiendo datos, que es lo opuesto a leer datos, en el puerto.

El anterior conjunto de niveles electrónicos en las líneas de direcciones IORQ es único para el chip I/O particular que está siendo escrito. Estos niveles son dispuestos en las líneas durante una fracción de segundo cuando se ejecuta el programa y, por consiguiente, deben ser reconocidos y utilizados por algún circuito electrónico rápido cogiéndolos "al vuelo". Este rápido circuito está contenido dentro del bloque de decodificación de direcciones, que, cuando reconoce este único conjunto lógico, pone su salida a nivel bajo, con lo que se activa el chip I/O. El chip I/O activado entonces toma el byte del bus de datos que la CPU ha dejado allí mientras están presentes las condiciones de activación anteriores.

La Figura 3.11 muestra un diagrama de un chip típico que realiza esta decodificación del bus de direcciones. También muestra el diagrama de terminales. El circuito integrado es un 74LS138, que se denomina un *decodificador de 3 a 8 líneas*. Su misión consiste simplemente en convertir los ocho bytes posibles en sus tres líneas de entrada (I∅, I1 y I2), en una salida (baja) en una de sus ocho salidas. La tabla lógica, o tabla de verdad, muestra cómo las entradas son convertidas a estados de salida por el CI. Por ejemplo, si las tres entradas son ceros, entonces la salida ∅ (terminal 15) estará activada (baja). Muchos de los circuitos lógicos que manejaremos tienen el estado bajo como el activo, lo cual, como ya hemos visto, se representa en los esquemas eléctricos por medio de una barra (negación) encima de la letra o letras con las que se denominan las líneas y terminales. De la misma forma, el valor decimal del número binario de entrada determina qué salida numerada se hace baja. Nótese también que esto sólo ocurre cuando el terminal 6 (Enable activo con nivel alto) tiene un uno, y los demás Enables, los terminales 4 y 5, bajos. Si cualquiera de los Enables está en estado equivocado, las salidas permanecerán altas sea cual fuese el estado de las entradas.

Como ejemplo, la Figura 3.11 muestra las líneas de direcciones A5, A6 y A7 conectadas en el orden correcto a las entradas y la línea IORQ a los dos Enables activos con nivel bajo. El Enable activo con nivel alto del terminal 6 se mantiene permanentemente alto, para que la activación del 74LS138 dependa exclusivamente de IORQ. Con esta explicación se pretende esclarecer que este circuito puede ser utilizado para activar un circuito I/O, cuando el bus de direcciones



Ē=0 y E=1

I2	I1	I0	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

TABLA DE VERDAD MOSTRANDO EL ESTADO DE SALIDA DEL 74LS138 PARA CADA ESTADO DE ENTRADA POSIBLE

Fig. 3.11. Decodificación

tenga un byte dado y el IORQ sea bajo del mismo modo que si se tratase de una sentencia IN o OUT. Si deseamos activar, digamos, el puerto 31 como anteriormente, A5, A6 y A7 serán bajos y, cuando IORQ sea también bajo, la salida numerada como Ø (terminal 15), se hará baja. Como ya hemos dicho anteriormente, esto sólo ocurre durante un tiempo muy corto, el que tarda la CPU en ejecutar el trozo de programa correspondiente. La electrónica que se conecte al 74LS138 debe ser capaz de capturar el contenido del bus de datos en ese tiempo. Sin embargo, esto no suele ser un problema, pues normalmente todos los CIs son más rápidos que la CPU.

En la Figura 3.10 está dibujado el tipo de CI de salida que debe ser conectado al decodificador de la Figura 3.11. Con las líneas de dirección mostradas, cada salida del 74LS138 es seleccionada para una dirección determinada en el bus de direcciones. Los chips I/O conectados a esas salidas, en cambio, pueden tener solamente un registro. Si uno de ellos tuviera dos registros, la Figura 3.12 muestra

cómo debe ser conectado. Habrá una entrada de direcciones en cada uno de los chips, quizá llamada C/D, como en la figura, que seleccione los registros de control y datos, respectivamente con 1 o cero. Esto se hace utilizando A5. La entrada I2 del decodificador puede mantenerse permanentemente baja y entonces solamente la mitad superior de la tabla de la figura 3.11 es activa. Habrá cuatro salidas decodificadas para esta situación, y cada una tendrá uno de esos chips conectados. A5 se conecta al terminal C/D de cada uno de los cuatro chips. Utilizando la tabla de la figura 3.11, y el hecho de que las líneas A∅-A4 son todas 1 para el Spectrum, podremos, trabajando con los bytes en el bus de direcciones, ver cuál activará esos chips y, por consiguiente, llevar a cabo el direccionamiento de sus puertos.

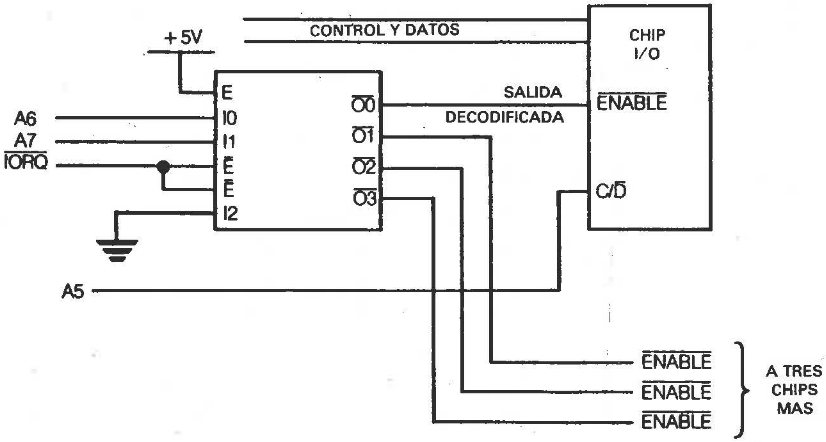


Fig. 3.12. Chip I/O de dos registros

Para añadir dos PIOs al Spectrum, cada uno de ellos ocupando 4 localizaciones de memoria, el 74LS138 debería tener I2 e I1 conectados a tierra, mientras que IO debería conectarse a A7, A6 y A5, y las demás líneas del bus de datos y control, conectadas a cada PIO como se muestra en la Figura 3.2. El terminal E-barra de uno de los PIOs deberá ser conectado a la salida ∅ (terminal 15) del 74LS138, y el otro a la salida 1 (terminal 14).

Esto nos da una idea del tipo de circuito requerido para periféricos generales para el Spectrum o cualquier otra máquina. Con objeto de captar mejor este aspecto de la técnica de computadores, podemos comenzar a interesarnos por aprender algo más sobre lógica electrónica. Hay muchos libros sobre la materia, de niveles

diferentes, desde los dirigidos al novato hasta el de ingeniero. Sin embargo, uno de los más didácticos y a la vez libro de referencia en esta materia es el *Recetario TTL*, de Don Lancaster, editado por Sams. Este libro se puede adquirir en la seguridad de que constituye un buen maestro y una fuente de información muy útil con la que podremos seguir el camino ya emprendido.

4
Periféricos robóticos

Introducción

El primer capítulo comenzaba con la descripción del significado de la robótica, definida como algunos aspectos de la inteligencia que los robots pueden poseer, refiriéndonos a las correspondientes cualidades humanas. En el capítulo anterior se presentaron, por medio del ratón robot, los principios básicos del control de la mecánica de un robot.

Este capítulo describe algunos interfaces importantes para utilización robótica en general, y pretende proporcionar, con el estilo de un recetario de cocina, la información que nos permita diseñar nuestros propios sistemas. Los periféricos introducidos aquí serán unas excelentes ayudas para los brazos de robot del capítulo siguiente, o bien para el ratón robot del capítulo anterior.

Cuando utilicemos un robot lo mejor es manejar cuanta más realimentación podamos. Hay muchos tipos de sensores en el mercado, algunos de los cuales proporcionan una tensión continua que varía de acuerdo con el parámetro físico medido. En este capítulo veremos cómo leerlos y utilizarlos en un robot.

Los movimientos son las salidas más importantes de un robot, utilizándose en muchos de ellos un tipo especial de motor, llamado motor *paso a paso*. Veremos un interface para utilizar el motor paso a paso en el diseño de máquinas móviles de cualquier género.

Control de movimiento

Como veremos en el siguiente capítulo, los brazos robots son un conjunto de palancas y juntas mecánicas, generalmente con una herramienta prensil en un extremo y una base firme en el otro. Las juntas son similares a nuestras propias articulaciones del brazo, y

tienen en su conjunto 4 ó 5 grados de movimiento o "grados de libertad". El brazo entero puede girar en círculo alrededor de la base, "codear" y "muñequear". Todas estas articulaciones proporcionan al brazo 4 grados de libertad, cuyos movimientos son continuamente variables o analógicos. El quinto grado podría ser el movimiento de la mano prensil, que puede estar simplemente abierta o cerrada, consiguiéndose su control por medio de un sólo interruptor.

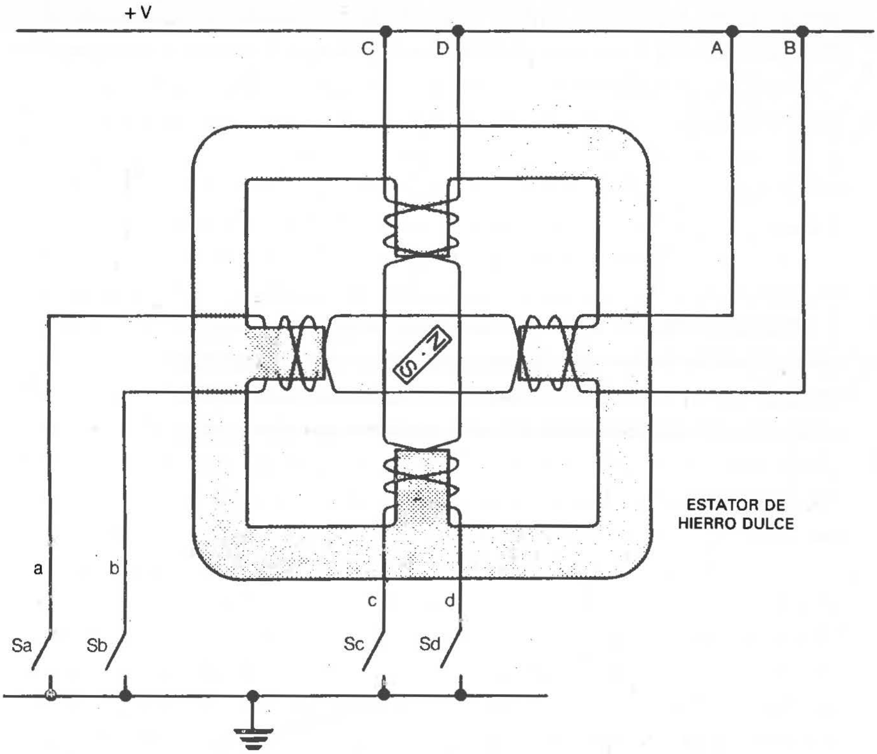
Hay varias maneras de crear estos movimientos, pero las más empleadas son las basadas en los fenómenos electromagnéticos. En el Capítulo 1 vimos como podían utilizarse el solenoide electromagnético y el motor de corriente continua. La rotación de un motor puede convertirse, por medio de palancas, manivelas o engranajes, en movimientos cualesquiera, y el control de estos movimientos dependerá forzosamente del tipo de motor y mecanismos usados. Uno de los motores más normalmente utilizados es el motor paso a paso que se describe a continuación.

El motor paso a paso

Como hemos visto, en los sistemas anteriores se deben producir movimientos rotatorios de precisión. Para conseguir esta precisión, sería utilísimo que dispusiéramos de un motor que no girara de una manera continua simplemente cuando se conecta a tensión, sino que lo hiciera de forma discontinua, por medio de giros discretos con un cierto ángulo, lo que lo haría fácilmente controlable. Este motor sería conectado a los mecanismos y, teóricamente, no necesitaría un sistema de control con realimentación para pararlo cuando llegara al final de su recorrido. Este motor ideal, que existe y se denomina motor paso a paso, se ha desarrollado justamente para facilitar el control por medio de sistemas digitales.

La utilización de un motor de este tipo facilita efectivamente los sistemas de control que pueden emplearse para el diseño de nuestros propios brazos robots, con la ayuda de un chip controlador normal, como veremos a continuación, para realizar más correctamente la electrónica necesaria.

Existen varios tipos de motores paso a paso y varios métodos de control. El tipo de motor que hemos elegido aquí como ejemplo para su descripción es el más sencillo de utilizar, que es el llamado motor paso a paso de *cuatro fases*. La figura 4.1 muestra una representación esquemática del motor. Hay detalles muy concretos del bobinado que pueden diferir de un motor a otro, aunque el de la



INTERRUPTORES CERRADOS	Sa & Sc	Sa & Sd	Sb & Sd	Sb & Sc	Sa & Sc
POSICION DEL ROTOR	N	N	N	N	N

Fig. 4.1. El motor paso a paso.

figura es uno de los posibles. El motor tiene tres partes principales: un imán pivotante en el centro llamado *rotor*, un *estator* de hierro dulce y cuatro bobinas arrolladas a dicho estator. El rotor magnético está rígidamente unido al eje de salida, que es el que se conecta a la maquinaria bajo control.

Para ver cómo están arrolladas las bobinas, fijémonos en una de ellas, por ejemplo en la bobina Cc. Empieza el bobinado en el saliente superior del estator, de arriba abajo, y continúa, también de arriba abajo, en el saliente inferior del estator, estando las vueltas

dadas en *la misma dirección*, para salir al exterior a continuación. Esto significa que si una corriente pasa desde C hasta c, cuando se cierra el interruptor Sc, los dos salientes del estator formarán una pareja de imanes eléctricos alineados en la misma dirección. Con ello se consigue que el rotor sea atraído, de manera que girará para alinearse con dichos salientes, con el polo norte del imán hacia arriba y el polo sur hacia abajo. La bobina Dd está arrollada en los mismos salientes, aunque en direcciones opuestas. Por ello, cuando pase la corriente por ella, el polo norte será atraído hacia abajo y el sur hacia arriba, en dirección opuesta. De la misma forma, la circulación de corriente por el bobinado Aa hará que el polo norte se oriente hacia la derecha y el Bb hará que se desplace hacia la izquierda. Controlando los interruptores de las cuatro bobinas, el rotor girará a voluntad tantos pasos como se desee y en la dirección más conveniente. Aunque existen limitaciones, éste es el principio fundamental utilizado para conseguir precisión en el movimiento.

Hay varios métodos posibles para controlar la dirección y posición del movimiento del rotor. En la tabla de la figura 4.1 podemos ver la secuencia para ir llevando al rotor a cuatro posiciones que forman ángulos rectos y en el sentido de las agujas del reloj. Nótese que este método requiere que haya siempre dos bobinados bajo tensión y que el imán se orienta a mitad de camino entre los dos campos magnéticos creados por ellos. Si se quiere invertir el sentido de giro del motor, bastará invertir el orden de dicha secuencia.

El motor ilustrado es capaz de girar en pasos de 45 grados si se quiere más precisión. Para ello habrá que añadir pasos de manera que el rotor se oriente, tanto en la dirección de los salientes del estator como en la mitad del camino entre ellos, manejando convenientemente los interruptores. A este modo de funcionamiento se le denomina *modo de medio paso.* Este método proporciona el mínimo ángulo de giro posible con cada motor. Los motores, en la práctica, tienen más salientes y, por tanto, más polos magnéticos en el rotor con objeto de reducir más el ángulo de paso. Un motor típico comercial tiene pasos de 7,5 grados o de 1,8, y la mitad en modo medio paso.

Los interruptores podrán cerrarse a mano utilizando relés, aunque ninguno de estos métodos es suficientemente rápido para el control robótico, por lo que se utilizan con preferencia interruptores a transistores controlados por líneas I/O en paralelo (PIO) de algún ordenador. Mediante un programa, se envían los impulsos necesarios en cada momento a los interruptores adecuados y se cuentan estos impulsos, para saber en qué situación se encuentra el meca-

nismo, según el tipo y número de impulsos enviados a las líneas de control.

Teóricamente, no es necesario disponer de realimentación en un proceso de control de este tipo, puesto que el ordenador debe conocer la posición de los mecanismos en cada momento, almacenando las órdenes ya enviadas. Sin embargo, puede ocurrir que el motor, en algún momento, realice giros que no debía, o no realice otros para los que se han enviado las órdenes oportunas, cuando existan condiciones especiales de carga o por otros motivos, por lo cual el ordenador tendría una idea falsa o desfasada de la posición real del mecanismo. Por ello, no es mala práctica incorporar en los elementos robóticos microinterruptores "fin de carrera" para informar al ordenador de algunas posiciones fundamentales para el funcionamiento del robot, como, por ejemplo, las posiciones iniciales o finales de una palanca. A veces es necesario informar al ordenador de todos los pasos para asegurar que cada orden tiene el efecto esperado. Este, claro está, es el sistema de más precisión.

Un proyecto con motor paso a paso

Con la idea de controlar el motor anteriormente descrito, debemos construir un circuito con interruptores a transistores, los cuales puedan ser controlados individualmente. Con el tipo de circuito empleado en el capítulo anterior pueden ser controlados pequeños motores paso a paso, con sólo disponer de transistores más potentes que los empleados allí para controlar relés. Necesitaremos diodos para proteger a los transistores, al igual que ocurría con el control de relés. Conviene que la alimentación sea tomada de una fuente distinta de la de los circuitos lógicos, que también requerirá condensadores de gran valor para el filtrado de ruidos eléctricos que puedan interferir con el procesador y la lógica electrónica.

El inconveniente con este tipo de control es que el programa BASIC puede no ser suficientemente rápido para obtener las mejores prestaciones del motor. Para realizar el control individual de las fases del motor, si se quiere que se haga de una manera correcta, son necesarias una cantidad considerable de instrucciones. Para llevar a cabo un control de este tipo de una manera flexible, probablemente tendríamos que utilizar instrucciones en código máquina, para conseguir la velocidad de proceso conveniente. Recomendamos, pues, este método como la última solución para el control de los motores paso a paso, aunque en una primera etapa nos contentaremos con el siguiente método de control, que es más sencillo de realizar.

Existe un circuito integrado en el mercado muy utilizado, el SAA1Ø27, que posee tres líneas de control por comandos e incluye transistores para gobernar directamente el motor. No incorpora el modo "medio paso", pero se trata de un chip excelente y sencillo para utilizar en aplicaciones todo paso. El único inconveniente es que no acepta señales de control TTL (lógica transistor transistor). Como veremos, necesitaremos algunos transistores para cambiar los niveles de tensión de las señales TTL del PIO, con objeto de poder controlar este chip.

A continuación se describe un experimento para hacer funcionar un pequeño motor paso a paso. Debemos adquirir un motor paso a paso adecuado para trabajar en conjunción con el SAA1Ø27. De otro modo, deberemos recordar que este circuito puede proporcionar una corriente máxima de 35Ø miliamperios por fase, a 12 V aproximadamente para cuatro fases. Necesitaremos también identificar los hilos del motor que van a recibir los datos o señales de control.

La Figura 4.2 muestra el esquema del circuito que utilizaremos como interface del motor con el SAA1Ø27, y de éste con el PIO. Si en el motor no estuviesen identificados con los convenientes rótulos los hilos correspondientes a las distintas bobinas del mismo, tendremos que averiguar qué hilo es cada uno de ellos, utilizando un multímetro que pueda medir ohmios. Como dichos bobinados tienen una resistencia de pequeño valor, habrá que seleccionar el rango de medida más sensible y medir las distintas combinaciones de los 5 ó 6 hilos que nos encontraremos. A veces, los hilos correspondientes a los rotulados como A, B, C y D en la Figura 4.2, están agrupados físicamente, y otras veces están formando parejas, como en la Figura CD y AB, por ejemplo, que pueden estar soldados juntos. También estarán los hilos que hemos llamado a, b, c y d. Si no están rotulados mediremos resistencia de cada pareja de hilos. Veremos que hay dos valores de resistencia, uno doble que otro. Por ejemplo, midiendo entre d y c, tendremos doble indicación que si medimos entre d y C. Mediante este procedimiento podremos averiguar, por lo menos, qué hilos se corresponden con las fases y cuáles tienen que estar juntos y conectados a la línea de +12 V, pero no podremos saber de qué fase se trata. Para averiguarlo, no tenemos más remedio que utilizar un procedimiento de pruebas y errores. Conectaremos el motor con cada combinación de hilos correspondiente a las fases, hasta dar con la combinación que lo haga funcionar correctamente, sin miedo a estropearlo si la conexión no es la correcta.

La Figura 4.2 muestra un circuito para controlar un motor, con

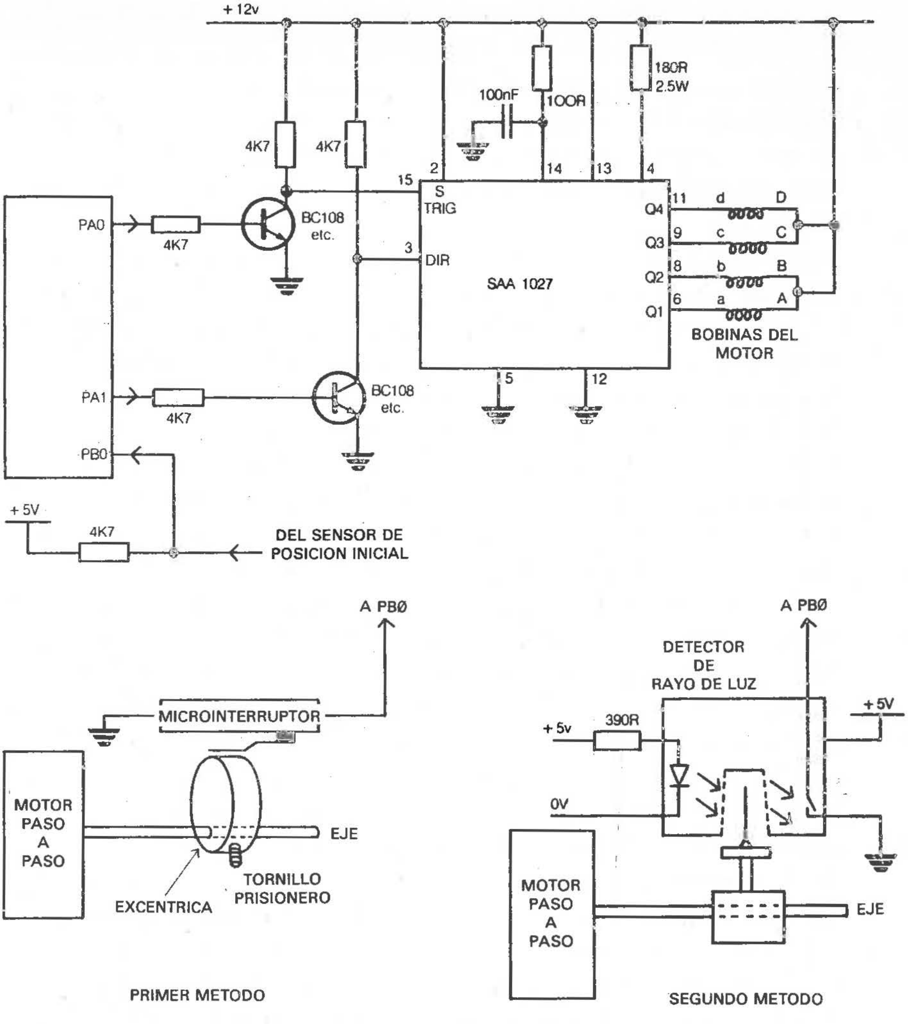


Fig. 4.2. Control del motor paso a paso.

realimentación desde un sensor de posición inicial. Como se dijo anteriormente, en condiciones de sobrecarga en el eje del motor, pueden producirse deslizamientos indeseados, en el movimiento del mismo, que pueden producir la pérdida de alguno de los pasos. Para prevenir esta posibilidad, y para hacer que el motor comience desde una posición conocida, puede incorporarse un sensor en el eje que permita identificar una posición dada como la posición inicial. El motor puede ser siempre llevado hacia atrás los pasos necesarios para encontrar dicha posición inicial por el ordenador y reiniciar el proceso de una manera correcta. El propio ordenador, al pasar el

mecanismo por esta posición inicial, tendrá una información de ello y verificará si ese momento coincide con el calculado por él mismo. Además, al comenzar el proceso, se puede conseguir fácilmente que se haga desde la posición adecuada sin más que contar los pasos necesarios desde esta posición de control, dependiendo de la aplicación mecánica en particular para la que se ha necesitado el motor paso a paso.

El PIO se utiliza como interface entre el motor y el sensor de posición inicial. Las salidas del PIO deben adaptarse para los niveles de +12 V compatibles con el SAA1Ø27, con transistores como los dibujados, del mismo tipo que los transistores del Capítulo 2. El SAA1Ø27 y el motor paso a paso se alimentan a partir de la misma fuente de +12 V. Todo esto se explicará en breve.

En la figura 4.2 se muestran dos tipos de construcción de sensores de la situación inicial. El primero de ellos está compuesto por un microinterruptor accionado por una leva colocada en el eje del motor. Se pueden encontrar dispositivos parecidos a éste en los comercios dedicados a artículos de modelismo, con un tornillo prisionero para fijar la leva al eje. Como alternativa, se puede usar un disco al cual se le suelde una palanquita. El microinterruptor deberá montarse de tal manera que sea accionado a cada revolución sin afectar al movimiento. Otro método de construir una leva es hacer una perforación excéntrica en un disco y pasar el eje a su través; mediante otro taladro lateral se podrá hacer pasar un tornillo sin cabeza para fijar el disco al eje. El microinterruptor debe montarse de tal manera que permanezca normalmente abierto, de manera que, cuando se cierre, proporcione una tierra (Ø V) a la línea de señal del PIO, que indicará qué motor ha llegado a la posición inicial.

También se muestra un tipo de sensor de posición inicial que no necesita ningún contacto mecánico ni eléctrico y que puede comprarse completo. Consiste en una caja de plástico provista de una ranura a través de la cual se dirige la luz de un LED hacia un detector luminoso. Este detector enviará su señal a un circuito lógico interno, que deberá mantener alta la señal de salida mientras la luz está llegando al detector. Mediante una leva colocada en el eje, se interceptará el rayo luminoso, haciéndose baja la salida, detectando de este modo la posición inicial. También puede adquirirse este tipo de interruptor luminoso sin la lógica interna, y realizar nosotros mismos un circuito que realice esta función.

Podemos tener nuestras propias ideas para realizar sensores de posición inicial y aplicarlas con idea de experimentar y producir sistemas más eficaces.

Fuente de +12 V

La fuente de alimentación del Spectrum no está diseñada para controlar o alimentar equipamientos mecánicos y, por ello, éstos deben ser alimentados por medio de una fuente de alimentación separada. La Figura 4.3 muestra un par de esquemas de circuitos para una posible fuente de alimentación, que dará una tensión sin regular para el controlador del motor. Si deseamos utilizarla como fuente primaria de los +5 V, tendremos que conectar simplemente los +12 V al regulador, como se describió en capítulos anteriores.

El principal elemento de una fuente de alimentación que extrae la energía de la red de 220 V es el transformador. El transformador

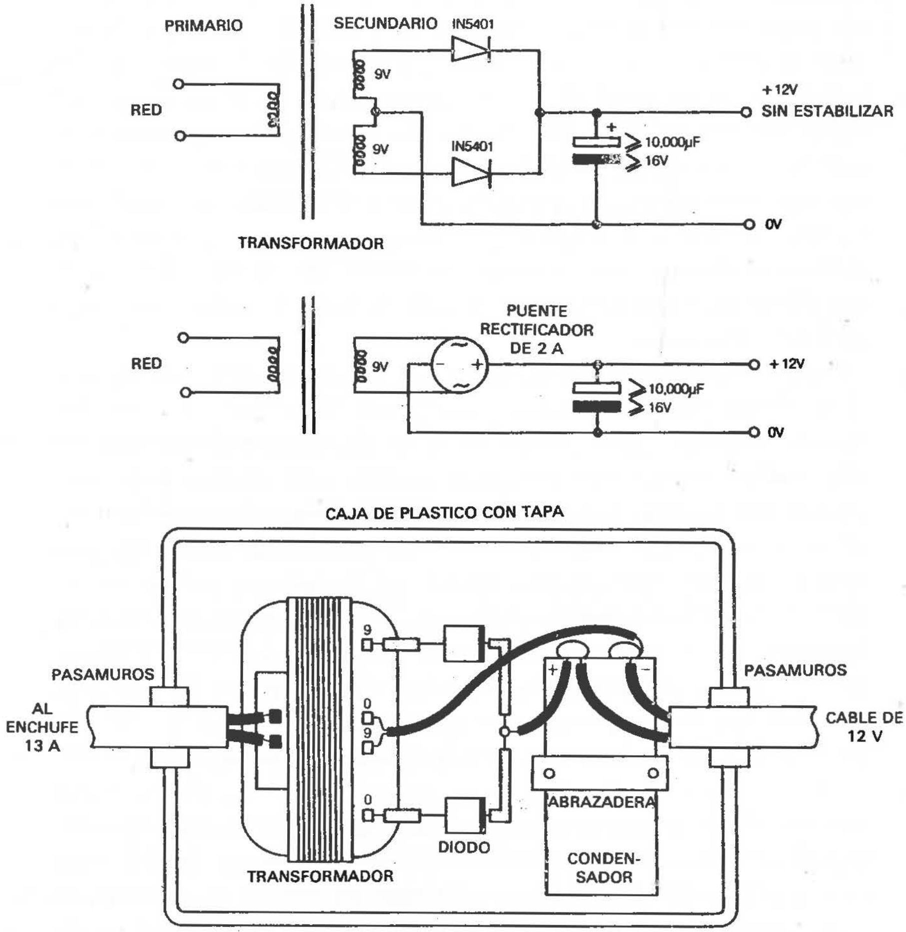


Fig. 4.3. Fuente de alimentación de +12 V.

es un componente pesado, que tiene un núcleo de hierro de una cierta forma al cual están arrollados dos o más bobinados. El objetivo es cambiar el voltaje que llega de la red a un valor más manejable y menos peligroso, con el que trabajan los circuitos electrónicos que estamos utilizando. El principio en que se basa es el siguiente: las fluctuaciones de tensión en el arrollamiento primario (que existen al tratarse de corriente alterna), producen variaciones de flujo magnético en el núcleo que, a su vez, inducen una tensión en el arrollamiento secundario de la misma forma que la primitiva, pero con valores relativos que guardan la misma relación que la que hay entre el número de vueltas del hilo en uno y otro arrollamiento.

Los voltajes proporcionados por los arrollamientos secundarios del transformador se rectifican por medio de diodos. Con un transformador que proporcione una tensión secundaria de 9 voltios eficaces, podremos, mediante su rectificación, obtener unos 11 V de tensión continua, que es el valor que necesitamos para nuestra aplicación. Necesitaremos que el transformador proporcione alrededor de 2 amperios para tener capacidad para dos o tres motores. Si utilizamos un transformador de un solo secundario de 9 V y 2 A, entonces utilizaremos el segundo circuito. En cambio, si utilizamos un transformador de 2 secundarios de 9 V y 2 amperios, será mejor utilizar el primero. Es lo mismo emplear una fuente con mayor capacidad de suministro de corriente, aunque la tensión de salida debe ser la correcta.

En el primer esquema se emplean dos diodos separados, capaces de rectificar grandes corrientes, con los dos arrollamientos secundarios del transformador puestos en serie. Es importante conectarlos exactamente como indica la figura. Para que los arrollamientos estén correctamente conectados en serie se debe respetar el sentido de las vueltas en el núcleo, pues si no, no obtendríamos ninguna tensión de salida. Puede que ya estén conectados en serie al adquirirlo, o puede ser también que estén los hilos separados, aunque rotulados los correspondientes a cada devanado con "Ø" y "9". En el primer caso no existe ningún problema. Para el segundo, tendremos que unir uno de los que están rotulados con "Ø", con el "9" del otro devanado, para realizar la conexión central de la línea de Ø V.

En el segundo esquema, en el que se utiliza un transformador con un solo secundario, se emplea un "puente" de rectificadores de algo más de 2 A. No es necesario —y pudiera, además, resultar muy caro y de grandes dimensiones— utilizar puentes de mucho más de 2 A, aunque el funcionamiento sería el mismo.

El condensador dibujado deberá ser del orden de 10.000 μF, con

una tensión de trabajo de 16 V o mayor. Probablemente será del tipo de conexión con terminales y tornillos. Debemos encontrar, en este caso, los terminales adecuados que nos permitan hacer la conexión con facilidad. Este condensador tiene polaridad, es decir, no pueden invertirse las dos conexiones. Un montaje incorrecto puede dar origen a una explosión. Algo parecido ocurre con el transformador con los devanados primario y secundario. Deberemos asegurarnos que el primario es el que va conectado a la red de 22Ø V.

Todo el circuito de la fuente de alimentación deberá ir montado dentro de una caja de plástico, con todo el cableado interno bien aislado y no precisamente con cinta aislante. Para aislar conexiones de hilo originalmente desnudo, se empleará macarrón de plástico, que puede conseguirse en establecimientos de venta de componentes electrónicos. Si no, se recurrirá a recuperar trozos de envoltura de cables. Es muy importante la perfecta realización de la fuente de alimentación. Muchas de las averías de los equipos electrónicos, en general, son, a menudo, producidas por causa de la fuente.

Tanto el transformador como el condensador deberán colocarse boca abajo en la base de la caja de plástico, utilizándose para su fijación, a ser posible, tornillos de plástico, para evitar cualquier aparición no deseada de tensiones peligrosas en el exterior de la caja.

La caja deberá cerrarse mediante una tapa de plástico bien segura. Los cables que salen al exterior deberán hacerlo a través de los correspondientes pasamuros. Utilizaremos una disposición como la mostrada en la figura 4.3, en la que vemos que la tensión de red está físicamente separada de las bajas tensiones. Insistiremos en que cualquier tipo de fijación de los componentes a la caja deberá ser de nylon para evitar la aparición de tensiones peligrosas en el exterior.

Finalmente, recordaremos que la tensión procedente de la red es peligrosa, por lo que la caja *nunca* deberá abrirse cuando el enchufe esté conectado a la red, ni aun en el caso de que el dispositivo esté apagado. Para mayor seguridad, emplearemos un fusible de 3 A en el propio enchufe.

Utilización del motor paso a paso

La manera en que vamos a utilizar mecánicamente el motor y las cosas que se van a mover gracias a él son enteramente de nuestra incumbencia. A continuación, tendremos información suficiente que nos capacitará para utilizarlo y controlarlo de la manera que deseemos.

El SAA1Ø27 tiene tres entradas, denominadas S, TRIG y DIR. La entrada S se utiliza para detener el motor y llevarlo a una posición de partida. Esta entrada será ignorada en nuestra aplicación. No obstante, podemos experimentar con esta entrada si lo deseamos. Nos detendremos por ahora en las otras dos entradas, mostradas en la Figura 4.2. Una de ellas, TRIG, es el control de paso. Para que pueda ser disparado el arranque del motor debe encontrarse a Ø V y, al recibir un impulso de subida (de Ø a 1), se moverá el motor un paso. Si en vez de un impulso recibe un tren de ellos, el motor se moverá continuamente. La otra línea de control, la DIR, es la que controla la dirección. El nivel de ésta determina la dirección de rotación en cada paso. No hay más controles que nos puedan preocupar. Con éstos, el SAA1Ø27 será capaz de atender al motor si se le envía la secuencia correcta.

Para realizar el control, el PIO deberá disponerse con las líneas PAØ y PA1 como salidas, y PBØ como entrada. Se podrían haber empleado otras combinaciones de las líneas del PIO, pero con ésta que indicamos resulta fácil la programación. Teclearemos:

```
OUT 95,255
OUT 95,Ø
OUT 127,255
OUT 127,1
```

El siguiente paso deberá ser enviar las líneas de control a un estado conocido, poniendo ambas líneas en estado alto, utilizando:

```
OUT 31,3
```

La verdadera dirección del motor puede ahora ser determinada y anotada. Para hacerlo, el motor debe moverse unas cuantas veces poniendo PAØ a nivel bajo y alto, alternativamente, mediante una sucesión de pulsos. El movimiento se origina cuando PAØ pasa del nivel alto al bajo, ya que el transistor invierte la polaridad del impulso. Lo haremos utilizando:

```
OUT 31,2
```

Estas instrucciones se han elegido para asegurar que PA1 permanece alta, esto es, dejando constante la dirección. Repitiendo las sentencias anteriores, haremos que el motor gire. Si no ocurre así, lo más probable es que esté mal conectado, con lo que tendremos que revisar las conexiones, según explicamos anteriormente.

A partir de ahora, deberíamos ser capaces de poder cambiar la dirección y repetir el proceso.

Para utilizar el sensor de posición inicial se deberá leer el estado de PBØ después de cada movimiento. Cuando la sentencia IN nos devuelva un número decimal par, sabremos que el equivalente binario de este número tiene a cero el bit Ø. En cambio, si el número que aparece es impar, entonces el bit Ø es 1. Si no nos resulta muy claro este razonamiento, volvamos a leer el Apéndice 1. Con esto tendremos un posible método de verificar este bit. El programa siguiente comienza disponiendo el PIO, para, a continuación, indicarnos que introduzcamos un comando. Así puede elegirse una de las tres opciones: colocar el motor en una posición inicial determinada, que dé un paso hacia adelante, o que dé un paso hacia atrás. Luego pregunta por el número de pasos requeridos. Posteriormente, si el sensor de posición inicial da un resultado positivo durante la ejecución de un comando de posicionamiento, aparecerá en la pantalla la palabra "POSICIONADO". Se asume que PBØ se hace bajo en la condición de posicionado.

```
 10 REM ** DISPOSICION DEL PIO
 20 OUT 95,255: OUT 95,0
 30 OUT 127,255: OUT 127,1
 40 OUT 31,3
 50 REM ** INTRODUCIR COMANDO
 60 PRINT "INTRODUCIR H PARA POSICIONADO"
 70 PRINT "B PARA ATRAS"
 80 PRINT "Y F PARA DELANTE"
 90 PRINT : PRINT
100 INPUT A$
110 IF A$="H" THEN  GO TO 400
120 IF A$="B" THEN  GO TO 600
130 IF A$="F" THEN  GO TO 800
140 GO TO 100
400 REM ** PRUEBA DE POSICIONADO
410 LET K=IN 63
420 IF 2*INT (K/2)=K THEN  PRINT
    "POSICIONADO": GO TO 40
430 REM ** AVANZA UN PASO
440 OUT 31,2: OUT 31,3
450 REM ** VOLVER A PROBAR
460 GO TO 400
600 REM ** PONER DIR. ATRAS
610 OUT 31,1
620 PRINT : PRINT
```

```
630 INPUT "CUANTOS PASOS?",A
640 REM ** MOVER 'A' PASOS
650 FOR I=1 TO A
660 OUT 31,0: OUT 31,1
670 NEXT I
680 GO TO 40
800 REM ** PONER DIR. DELANTE
810 OUT 31,3
820 PRINT : PRINT
830 INPUT "CUANTOS PASOS?",A
840 REM ** MOVER 'A' PASOS
850 FOR I=1 TO A
860 OUT 31,2: OUT 31,3
870 NEXT I
880 GO TO 40
```

Observaremos que las velocidades de los pasos de giro son relativamente bajas y no pueden ser aumentadas utilizando el BASIC. Sin embargo, como ya dijimos, muchas de las funciones robóticas no deben ser necesariamente rápidas. Las velocidades pueden ser las adecuadas si se utiliza el motor adecuado con los mecanismos adecuados.

Uno de los problemas que nos encontraremos con el sensor de posición inicial es que pudiera dar indicación positiva no sólo para una posición, sino para varias posiciones muy juntas. Deberemos ser conscientes de este hecho y tenerlo en cuenta cuando confeccionemos nuestros programas de control, pues puede ser necesario más de un paso para alcanzar la posición deseada. Naturalmente, esta imprecisión dependerá de los mecanismos que tengamos. Por supuesto, lo ideal es ajustar el sistema mecánico lo mejor posible para tratar de conseguir encontrar la posición única.

Podemos hacer muchas mejoras a este programa, por ejemplo, se podrían almacenar los movimientos que se ordenan desde el exterior, mediante comandos, para poder repetir la misma secuencia cuando deseemos. Este es un ejemplo muy simple de las técnicas que pueden emplearse para aprendizaje de un robot.

El motor puede utilizarse para hacer que se mueva un ratón robot por el suelo; posicionando su rueda de dirección con gran precisión, podemos hacer que giren las manecillas de un reloj, mover una mano artificial, un brazo, etc. El motor paso a paso es un versátil y útil dispositivo que proporciona un ejemplo excelente de máquina para convertir comandos electrónicos en movimientos controlados.

El apartado siguiente contiene información de otro tipo de interface robótico, que nos permitirá experimentar con la consecución de implementar atributos humanos en máquinas, como es la síntesis de voz.

Síntesis de voz

Como se menciona en el Capítulo 1, hay dos tipos de sistemas de síntesis de voz. Este tipo que describimos aquí es el llamado de *voz almacenada,* y consiste en un chip procesador de voz (SPC) y una ROM de voces, que contiene los datos de las palabras que deben ser pronunciadas. El SPC es lo suficientemente inteligente como para tomar una palabra de 1 byte de un PIO, por ejemplo, y llevar a cabo la tarea completa de ir a buscar el dato de voz, y realizar la pronunciación que se le pide. La salida es una forma de onda de audio, que requiere una cierta cantidad de proceso y filtrado, antes de amplificarla. El chip SPC que hemos utilizado es el MM541Ø4 de la casa National Semiconductors. Las ROMs pueden adquirirse grabadas con vocabulario standard para este chip, con 5Ø ó 7Ø palabras cada una. Este chip es extraordinariamente fácil de utilizar y es un ejemplo muy representativo de interface de sistemas con el Spectrum. Existen otros procesadores de voz similares en el mercado, y la mayoría de ellos se utiliza de manera parecida.

La Figura 4.4 muestra un diagrama de bloques del interface sintetizador de voz, y la Figura 4.5 proporciona el diagrama del circuito completo. Para realizar el proyecto de sintetizador de voz, necesitaremos un SPC y, al menos, una ROM de voces con el vocabulario que elijamos de los existentes. En la Figura 4.5 aparece también el número de código del SPC, por si existiera alguna dificultad en su localización. Para elegir la ROM de voces tendremos que consultar un catálogo que nos permita seleccionar las palabras más convenientes de los diferentes vocabularios que se ofrecen. También conviene hacerse con las hojas de características de ambos circuitos integrados. Con ello dispondremos de detalles técnicos con los que poder realizar nuevos experimentos. Los demás componentes especificados en el proyecto son normales y se encuentran con facilidad.

Para explicar el funcionamiento del sistema, la figura 4.4 muestra el SPC conectado a un PIO y a un circuito de audio. La palabra que debe ser pronunciada se selecciona suministrando un conjunto de bits al SPC a través de sus líneas de datos de palabra. Cuando la línea WR del SPC pasa de Ø a 1, el SPC toma este conjunto de bits,

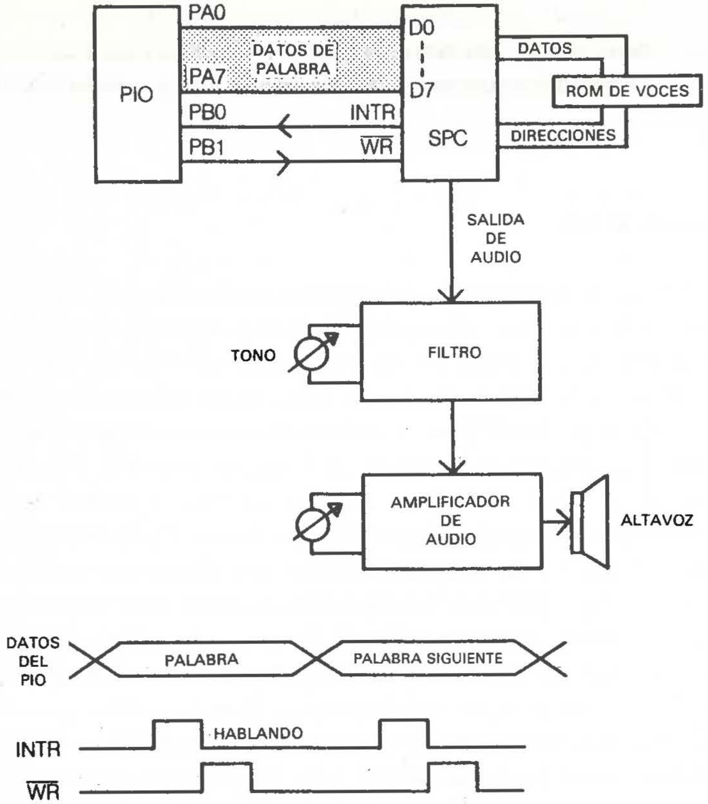


Fig. 4.4. Diagrama de bloques del sintetizador de voz.

consulta a la ROM y produce los sonidos de la palabra en su salida de audio.

Para evitar que el ordenador interrumpa este proceso, hay una salida del SPC que avisa al ordenador de que está ocupado. Esta línea se llama INTR (interrupción). Cuando el SPC está hablando, la línea INTR toma el valor cero; cuando acaba la pasa al valor 1. Así pues, esta línea proporciona realimentación al ordenador para que el programa procese a la velocidad justa para conseguir el mejor efecto. El ordenador leerá esta línea y esperará hasta que esté a nivel alto, antes de cambiar de nuevo el estado de WR. En la parte inferior de la figura se ha dibujado un diagrama de tiempos, con el que se pretende mostrar, más que los tiempos propiamente dichos, la secuencia de sucesos. Se dispone una palabra sobre las líneas de datos del PIO, se verifica el estado de INTR, espera el ordenador a que se ponga a nivel alto, la línea WR pasa de alta a baja y viceversa, y el SPC comienza a hablar. La línea INTR se hace baja para indicar

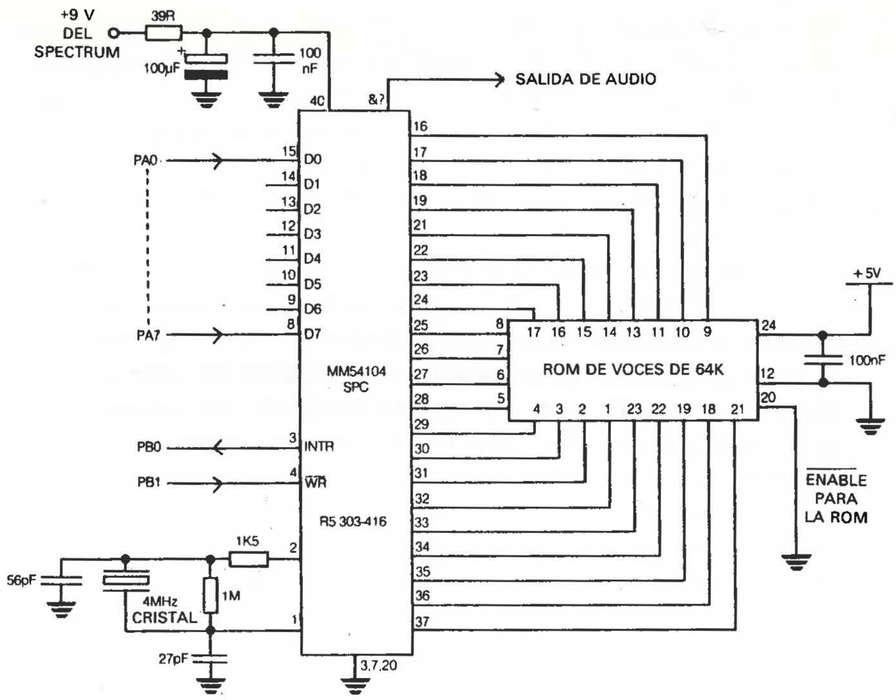


Fig. 4.5. Chip sintetizador de voz.

este estado. La siguiente palabra se puede disponer entonces en las líneas de datos y no ocurrirá nada hasta que el ordenador pase de nuevo WR a alto. Si la línea WR se hiciese alta mientras el SPC estuviera hablando, sería interrumpida la palabra y comenzaría la siguiente palabra a ser procesada.

Para una palabra dada, el SPC recoge todos los datos de la ROM de voces y busca todo lo necesario utilizando sus buses de datos y direcciones. Es esencial que la ROM que se compre sea compatible específicamente con el MM541Ø4; otro tipo de ROM no funcionará. Cuando los datos son recogidos, el SPC los convierte en una señal de audio muy parecida a la voz humana.

La ROM tiene una línea Enable (habilitación), la número 2Ø, como se ve en la Figura 4.5 que se utiliza para activar la ROM cuando se conecta a Ø V, permanentemente, si sólo se utiliza un chip de ROM. Sin embargo, si se necesita más de una ROM para aumentar el vocabulario, las líneas Enable de cada una se deberán conectar a las líneas de salida del PIO, con objeto de seleccionar en cada momento la ROM que se necesite, poniendo su entrada Enable a nivel bajo y desconectando las demás para evitar interferencias. Los

buses de direcciones y datos de las ROMs se conectan todas en paralelo al SPC. Si se quisiera aumentar aún más el número de chips de ROM que pueden ser direccionados directamente, habrá que usar un 74LS138 para decodificar 3 de las 6 líneas PB de reserva del PIO, como se explicó en el capítulo anterior. Mediante este procedimiento pueden habilitarse un total de 16 chips de ROM desde las 6 líneas PB del PIO.

La salida de audio del SPC proporcionará palabras reconocibles cuando se amplifiquen, pero dado que la fuente primaria de la señal de salida es una señal digital, aquélla contendrá componentes de frecuencia indeseadas que degradan la inteligibilidad. Las frecuencias requeridas están contenidas en una banda de, aproximadamente, quinientos o seiscientos kilohercios, que corresponde con la frecuencia del DO más alto del piano, aproximadamente. Para conseguir voz de buena calidad, además, necesitaremos filtrar esta banda de frecuencias y llevarla al amplificador de audio. Esto se realiza con un filtro electrónico, que puede incorporar un control de tono. El amplificador, con su control de volumen, es el eslabón final de la cadena que termina en el propio altavoz.

Electrónica del sintetizador de voz

La Figura 4.5 muestra el diagrama de conexión del SPC y la ROM. Como vemos, están literalmente pegados uno a otro, con lo que la construcción sobre un trozo de placa Veroboard se facilita bastante. La ROM necesita +5 V y el SPC +9 V. La fuente de alimentación de 5 V descrita en el capítulo 2 es suficiente para la ROM. Los 9 V provienen de la fuente de alimentación del Spectrum, saliendo por el conector posterior del mismo, y pasan a través de un filtro sencillo, resistencia-condensador, cuyo objeto es reducir el zumbido de la fuente de 9 V que puede oírse en el altavoz. El ruido electrónico se debe eliminar también de la fuente que alimenta a la ROM por medio de un condensador de 10 nF, colocado tan cerca como se pueda de la propia ROM.

Para que funcione el SPC también se requiere un reloj de 4 MHz, que se consigue conectando al SPC un cristal de cuarzo como se indica en la figura. Los componentes se consiguen todos fácilmente, incluso el cristal de cuarzo, por ser de una frecuencia muy empleada. Los condensadores son del tipo cerámico, de disco.

La Figura 4.6 muestra el circuito de audio. El filtro se realiza con un circuito integrado 741 dispuesto como filtro activo, y alimentado

con los +9 V del Spectrum más su propia fuente separada y filtrada. El 741 es del mismo tipo que el que se utilizó para el detector de luz del capítulo anterior, aunque en esta configuración de filtro activo necesitará alimentarse a partir de 9 V en vez de 5. Podemos experimentar cambiando algunos de los valores de los componentes mostrados en la figura hasta obtener los mejores resultados. Cambiando la resistencia de 22Ø K y el condensador de 1Ø nF que están conectados entre los terminales 2 y 6, varían las características del filtro y, por tanto, las características de la voz.

El amplificador de audio se ha realizado con un circuito integrado convencional, el LM38Ø. La versión que hemos utilizado es la de encapsulado de 14 terminales. Se puede utilizar cualquier otro amplificador, pero habrá que conocer la disposición y numeración

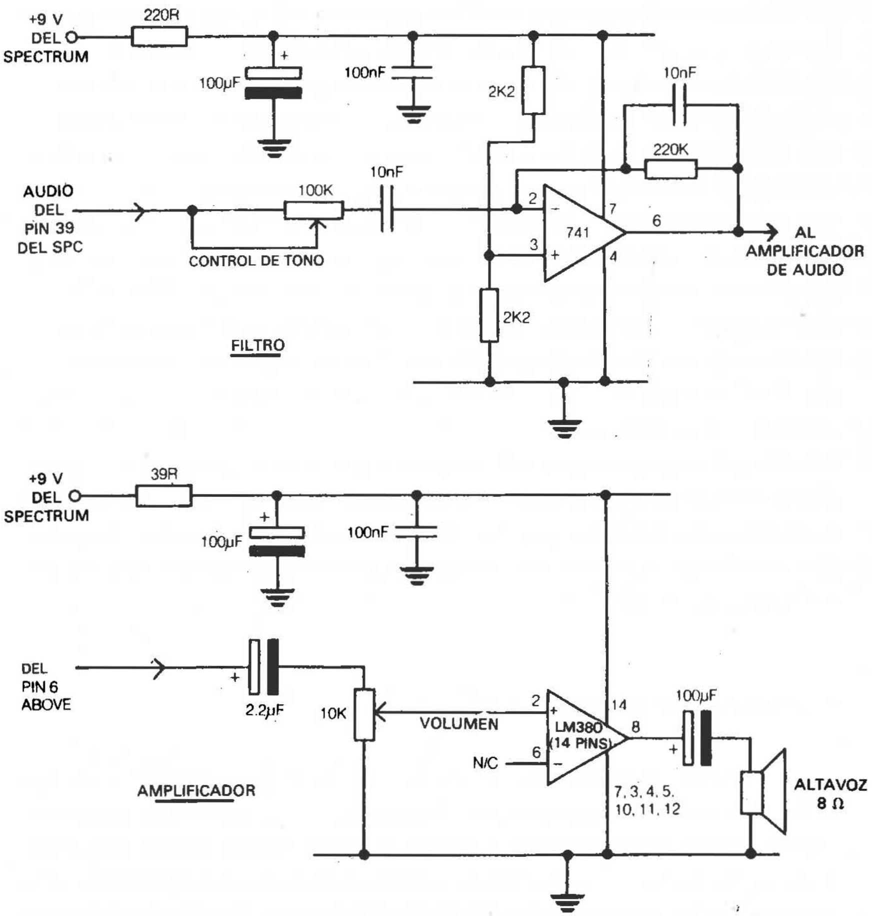


Fig. 4.6. Circuito de audio para el sintetizador de voz.

de sus terminales para conectarlo correctamente. Con este amplificador se consigue una potencia de salida de audio de unos 1,5 W, que es una potencia suficiente para nuestros propósitos. Se puede obtener más potencia de él, aumentando el voltaje hasta un máximo de 15 V, lo que produciría una potencia de hasta 2,5 W.

Los condensadores de la Figura 4.6 son todos cerámicos, de disco, excepto los electrolíticos. Estos deberán conectarse con su polaridad correcta. Deben poder soportar una tensión de trabajo de 16 V como mínimo. No es necesario utilizarlos con tensiones de trabajo mucho mayores, ya que, aunque funcionarían normalmente, serían de un tamaño demasiado voluminoso y haría el montaje poco limpio. En los circuitos de audio no deben usarse hilos de conexión largos, pero tampoco se deben poner demasiado juntos formando una especie de nido.

Si hubiese demasiado ruido o zumbido en la salida, deberemos intentar ajustar los valores de los componentes de filtro de ruido conectados a las líneas de alimentación de 9 V. Normalmente lo reduciremos, aumentando el valor del condensador y de la resistencia, pero ésta no debe aumentarse tanto que la tensión baje significativamente, de modo que el circuito pierda eficiencia.

La calidad que se obtiene de este circuito es bastante buena, si se considera la simplicidad de la electrónica utilizada. Todas las palabras serán comprensibles y contendrán incluso sonidos con inflexiones naturales. Naturalmente, puede mejorarse la calidad utilizando un circuito de filtro más sofisticado. En las hojas de datos técnicos del SPC se sugiere el uso de circuitos más compejos, que podemos utilizar si nos interesan.

Como experimento, intentemos saltarnos el circuito de filtro. Para ello se deberá tomar el terminal 39 del SPC directamente al lado positivo del condensador de entrada del amplificador de audio. Esto ilustrará el efecto del filtro, que procesa la señal de audio que proporciona el SPC.

Utilización del sintetizador de voz

La Tabla 3.1 proporciona los puertos I/O para el PIO. Con esta información y el diagrama de conexión de la figura 4.5, podemos hacer que un programa haga hablar al sintetizador, usando las palabras contenidas en la ROM de voces asociadas al Ø (decimal). Por ejemplo, si se ha elegido la ROM "Números y Letras", será pronunciada la palabra "uno'.

```
10 REM *** DISPONER A DEL PIO
20 REM *** COMO SALIDAS
30 OUT 95,255: OUT 95,0
40 REM *** DISPONER B DEL PIO
50 REM *** PB0 ENTRADA, RESTO SALIDAS
60 OUT 127,255: OUT 127,1
70 REM *** PONER TODAS LAS SALIDAS A 0
80 OUT 31,0: OUT 63,0
90 REM *** IMPULSO A LINEA WR
100  OUT 63,2: OUT 63,0
```

El método empleado para que el PIO seleccione las líneas I/O se explicó en el último capítulo. Esta rutina pone todas las líneas PA a cero, seleccionando la primera palabra en la ROM. Entonces se hará alta la línea WR conectada a PA1 y, a continuación, baja (impulso). El flanco de subida del impulso enviado a esta línea desencadenará la pronunciación de esta palabra "uno".

El programa anterior no utiliza el INTR, pues es sólo necesario para encadenar palabras formando frases. Su misión principal es que las palabras se pronuncien seguidas unas detrás de otras, sin que se solapen. Si en una aplicación particular no es imprescindible otra cosa, puede hacerse una pausa constante entre cada dos palabras, sin necesidad de utilizar la línea INTR. Por ejemplo, si el sintetizador está llamando a un conjunto de números, se puede utilizar una instrucción PAUSE entre cada número que haya de ser pronunciado y el siguiente, de tal manera que se consiga el suficiente retardo para evitar interferencias entre ellos.

Sin embargo, en el caso de tener que pronunciar una frase de una manera natural, la interrupción en el momento exacto es esencial, así como los intervalos de silencio. El siguiente programa hace que se pronuncien las primeras 1Ø palabras de la ROM, utilizando INTR para asegurarse que no hay solapamiento entre palabras y utilizando PAUSE para conseguir tiempos constantes entre palabras.

```
10 OUT 95,255: OUT 95,0
20 OUT 127,255: OUT 127,1
30 OUT 31,0: OUT 63,0
40 FOR I=0 TO 9
50 REM *** DISPONER PALABRA
60 OUT 31,I
70 REM *** COMIENZA VOZ
80 OUT 63,2: OUT 63,0
90 REM *** VERIFICAR INTR Y
```

```
100 REM *** CONT. SI ALTA
110 LET K=IN 63
120  IF K=(K/2)=K THEN  GO TO 110
140  PAUSE 10
150 NEXT I
```

Ahora debemos experimentar con rutinas escritas para encadenar palabras seguidas en forma inteligible.

Aplicaciones robóticas generales

La síntesis de voz es una de las cualidades que pueden añadirse a un robot, dándole personalidad y mayor habilidad para enviar información hacia el controlador. Por ejemplo, si un robot controlado por la luz descubre luz, podría decirlo, dirigirse hacia ella y seguir informando de lo que lleva recorrido mientras no la pierda de vista. Podemos añadir al robot detectores de choque y programarlo de manera que nos diga algo cuando encuentre un obstáculo. Podría, por ejemplo, pedir ayuda. Este modo de interacción es la base de la inteligencia de la máquina, y si una pantalla de televisor nos ha sido fundamental hasta ahora para recibir realimentación de la máquina, ahora podemos recibir la comunicación de la misma por medio de la voz también.

Cualquier sistema robótico podrá disponer también de la habilidad de monitorizar distintos niveles de un estímulo dado, que puede ser muy útil en muchas aplicaciones. Hasta ahora, sólo hemos utilizado impulsos digitales, pero en la naturaleza muchos impulsos son analógicos más que digitales, esto es, varían de valor de una manera continua, no sólo están o no están, o es una cosa o su contraria. Esto ocurre, por ejemplo, con estímulos tales como la temperatura, la presión, el ángulo de un brazo mecánico, etc. Hemos visto ya uno de estos detectores digitales, llamados "todo o nada". Con este detector se obtenía un 1 si la luz superaba un cierto umbral, o un cero si estaba debajo.

Con objeto de manejar este concepto de niveles continuos de un parámetro, primero lo convertiremos en niveles de tensión función de ellos. Esto es común a los detectores digitales; en el detector de luz ya visto, esta función la realizaba la LDR. Para la detección de otros parámetros se utilizan conversores o transductores de la variable física en tensión, como los termómetros electrónicos, por ejemplo. Una vez obtenido el voltaje dependiente del parámetro, debemos

convertir cada valor del mismo en información digital, de 8 bits cada uno, para que lo entienda el ordenador, leyendo estos datos en un bus de datos, por ejemplo. La conversión de parámetros analógicos en valores discretos, digitales, se denomina *conversión analógica a digital.* Existen convertidores de este tipo comerciales (ADC) contenidos en un circuito integrado. El apartado siguiente muestra la utilización que puede hacerse de uno de estos dispositivos y sugiere cómo, acoplándose una resistencia dependiente de la luz (LDR), se pueden monotorizar niveles variables de luminosidad.

Conversión analógico digital

La Figura 4.7 muestra un diagrama de bloques de un interface ADC. Una LDR, a la que se le ha llamado R2, cambia el valor de su resistencia de acuerdo con el nivel de luz que incide sobre ella. Cuanta más luz le llegue menor será su resistencia. Esto produce voltajes variables a la entrada del ADC. El ADC se encarga de comparar la tensión de entrada con una tensión de referencia y, según sea la diferencia, generará conjuntos de números binarios distintos. El PIO lee este número binario, con lo que desde el ordenador pueden seguirse los cambios de niveles de luz.

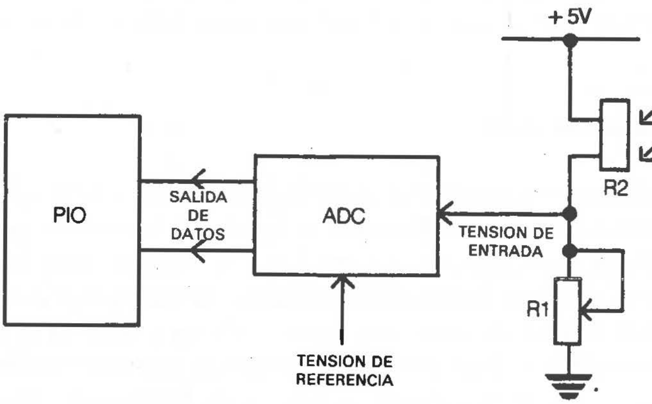


Fig. 4.7. Conversión analógico-digital.

Si el número binario producido tiene 8 bits, el ADC proporcionará 255 (decimal) cuando el voltaje de entrada sea igual al de referencia, y dará ∅ (decimal) cuando el voltaje de entrada sea cero. Así pues, los márgenes de variación de la tensión de entrada hay que

limitarlos entre ∅ V y la tensión de referencia del ADC. Por ejemplo, si la tensión de referencia del ADC es de +5 V, entonces limitaremos las tensiones de entrada entre ∅ y +5 V, con lo que dicho margen estará dividido en tantos niveles como números puedan formarse con ocho bits, es decir, 256. La diferencia de voltaje entre los distintos niveles será:

$$\frac{1}{256} * 5 \text{ V} = \emptyset{,}\emptyset 195 \text{ V}$$

Como los niveles de tensiones son discretos, puede ocurrir que no sean detectados por el ADC pequeñas variaciones de tensión. Solamente cuando la tensión supera uno de estos niveles, el número binario cambiará de valor, permaneciendo en este último hasta que se supera el siguiente nivel. La diferencia en tensión antes calculada determina la resolución o precisión del ADC. La teoría del ADC es muy compleja, y no la vamos a ver aquí. Nos limitaremos a presentarlo con un ejemplo práctico de su utilización.

La LDR de la figura 4.7 tiene una resistencia que puede variar desde cientos de kiloohmios a unos pocos cientos de ohmios, según cambie el nivel de luz. Es difícil elegir el valor adecuado de R1, por lo que lo mejor es utilizar en su lugar un potenciómetro dispuesto como resistencia variable de 1∅ K. Con ello podemos ajustar el mejor valor que se adapte a nuestras condiciones de iluminación.

Un ejemplo de ADC

La Figura 4.8 muestra un ejemplo de un circuito ADC que añade algunas cosas al circuito básico de la figura 4.7. El circuito integrado que hace la conversión es el ZN427, que es bastante fácil de conseguir, tiene un canal de entrada analógico y la salida digital de 8 bits. El voltaje interno de referencia es de 2,5 V, que aparece al exterior por el terminal 8. Este voltaje de referencia hay que conectarlo al terminal 7, que es la entrada de tensión de referencia exterior. La señal analógica, que hay que comparar con la de referencia, debe conectarse al terminal 6. Para asegurar que la referencia permanece estable, se conecta un condensador de 1 μF, entre la tensión de referencia y tierra, como se muestra en la figura. La resistencia de 39∅ ohmios hay que conectarla, siendo necesaria para que funcione correctamente el circuito de tensión de referencia. Nótese que hay un condensador de 1∅∅ nF entre el punto de +5 V (terminal 1∅) y tierra.

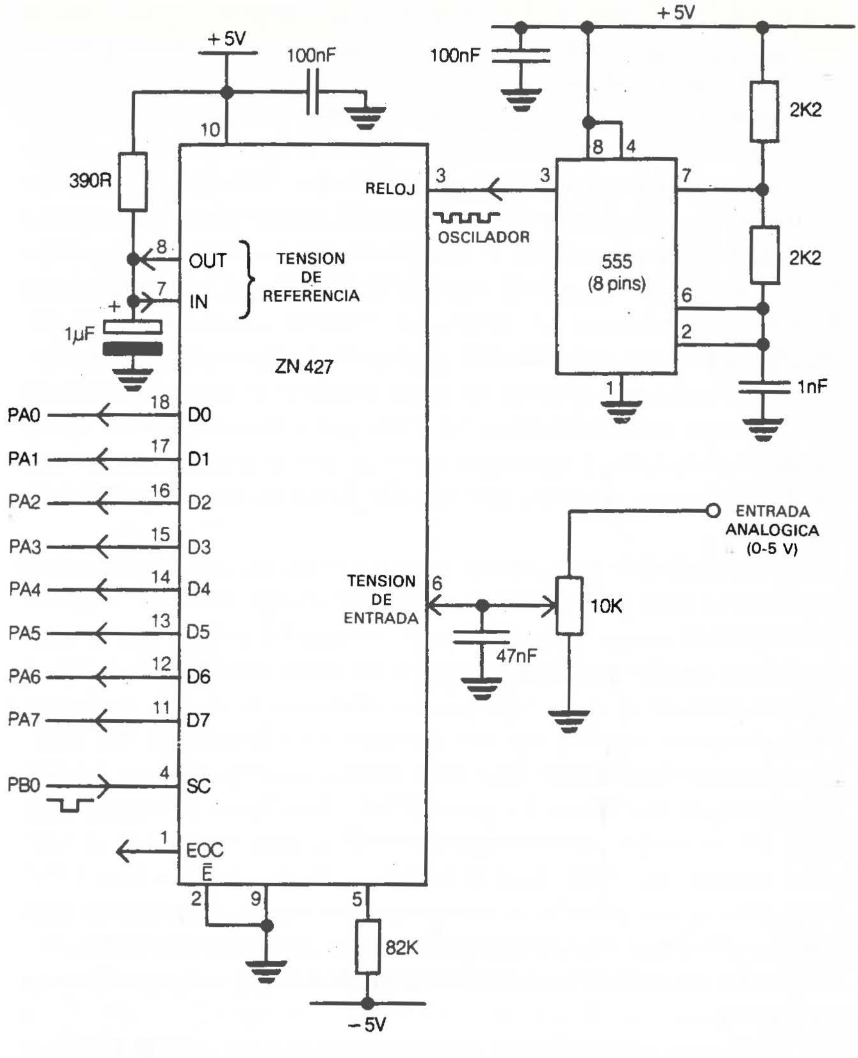


Fig. 4.8. Convertidor analógico-digital (ADC).

Su misión es la de reducir el ruido electrónico procedente de la fuente de alimentación.

El ZN427 también necesita una tensión de alimentación de —5 V. Se consigue por medio de una resistencia de 82 K conectada entre el terminal 5 del ADC y el conector posterior del Spectrum. El número binario de salida se envía directamente al PIO para poderlos utilizar por el ordenador.

Además de estos 8 bits, se necesita uno más para comandar el ADC y que realice la conversión. Se realiza el comando en el termi-

nal 4 del ZN427, que se denomina línea de *comienzo de conversión* (SC). En la figura aparece esta línea procedente de la línea PBØ del PIO, que para este caso se debe disponer como salida del PIO tal, que cuando se haga baja hará que comience la conversión, para lo cual el ZN427 tomará la tensión analógica de la entrada y la comparará con la de referencia, produciendo el número binario correspondiente. A continuación pondrá alta su línea EOC (fin de conversión) que aparece en el terminal 1. Esta línea puede utilizarse para interrupciones si se requiere. El número binario de salida permanecerá en las líneas de salida, sin variar, hasta que se produzca el siguiente flanco de bajada en la línea SC que iniciará una nueva conversión. En efecto, las líneas de datos de salida pueden colocarse en triestado tal como las líneas de datos del PIO, pero en este circuito se ha dejado baja la línea E-barra (terminal 2), que es la que controla esta posibilidad, para habilitar permanentemente las líneas de datos de salida.

El proceso de conversión es secuencial, por lo que se necesita un oscilador o reloj para temporizar las actividades del ADC. Aquí se ha empleado como reloj un circuito integrado, el 555, que es muy popular y puede utilizarse para generar gran variedad de relojes y temporizadores a muy bajo precio. Podríamos utilizar cualquier oscilador para este fin, con una frecuencia no superior a 9ØØ KHz. Esta frecuencia es varias veces más rápida que el programa BASIC que gobierna el proceso. Después de ocurrir el flanco de bajada en la línea SC, se tardan aproximadamente 45 μs para realizar la conversión, usando este reloj. Sólo es necesario monitorizar la línea EOC para saber que el proceso se completa y, si nuestro programa es muy rápido, producir una interrupción antes de enviar la orden SC. Como no es nuestro caso, entonces la línea EOC se ignora en esta aplicación.

El circuito integrado 555 tiene un condensador de 1ØØ nF en su línea de alimentación, para reducir el ruido que pueda provenir de otros dispositivos alimentados por la misma línea, y de la propia fuente. Este condensador cerámico de disco debe montarse muy cerca del 555 y sus terminales se deberán dejar lo más cortos posible.

La señal de entrada varía de Ø a 5 V, por lo que necesita ser reducida para que varíe entre Ø y 2,5 V, que es la tensión de referencia y, por tanto, el límite superior conveniente. Para ello se utiliza un potenciómetro de 1Ø K, que dividirá por dos las tensiones de entrada cuando el cursor esté en el punto medio. El condensador de 47 nF, también cerámico, de disco conectado a la entrada analógica del ADC, se utiliza para evitar picos de tensión a la entrada, que si se

producen cuando se está realizando la conversión, pueden falsear el proceso.

Con este sistema se obtiene una precisión teórica de 1 en 256, es decir, del 0,4%, que es más que suficiente. Si prescindimos del bit menos significativo (LSB), seguiremos teniendo una gran precisión, por ejemplo, del 1%. Conseguiremos con ello el poder utilizar la línea correspondiente del PIO para este bit, como línea para activar el SC, con lo que sólo utilizaremos un lado completo del PIO en lugar de usar un lado completo y el otro lado con una sola línea para este fin. En la Figura 4.9 se muestra esta sugerencia de interface. Todas las demás conexiones que no aparecen son las mismas que en la Figura 4.8. Se ha dibujado también la LDR con su potenciómetro de ajuste. Utilizaremos este circuito en adelante.

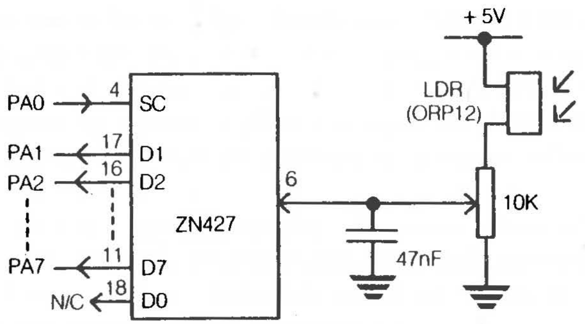


Fig. 4.9. Aplicación para el ADC.

El circuito completo lo construiremos en placa Veroboard, como es normal, y lo alimentaremos con la fuente descrita en el Capítulo 2. La fuente ZX de 9 V tiene capacidad suficiente para este proyecto. Antes de colocar los CIs en sus respectivos zócalos se debe alimentar el circuito, verificando si aparece el mensaje inicial en la pantalla al encender el Spectrum, y verificando asimismo si las tensiones de alimentación aparecen en los terminales correctos. También es una buena idea comprobar que la tensión en el terminal 6 del zócalo del ZN427 varía cuando la luz varía y no excede de 2,5 V en ese punto. Si es necesario, habrá que ajustar el potenciómetro para conseguirlo. Habiendo hecho estas pruebas, se desconectará el circuito de la alimentación, se colocarán los chips en sus respectivos zócalos y se volverán a realizar las pruebas anteriores, una vez alimentado de nuevo, teniendo cuidado para no cortocircuitar con la punta de prueba los terminales próximos de los CIs. Para ver si el 555 fun-

ciona se necesitan algunos medios auxiliares de prueba. Una punta de prueba lógica será suficiente para ello.

Si todo está correcto, podremos verificar la operación del ADC utilizando el programa del apartado siguiente. Esto nos permitirá utilizar el circuito para muchos tipos de realimentación analógica de nuestro robot.

Si necesitáramos más de un canal ADC para varias tensiones analógicas, deberemos colocar otro ZN427 al otro lado del PIO, si lo deseamos, y otro PIO con dos convertidores. Todos ellos podrían utilizar el mismo reloj proporcionado por un solo 555. Sin embargo, lo mejor para seguir experimentando es acopiar experiencias mediante el uso exhaustivo de ejemplos simples. Hay varios ADC multicanal en el mercado, cuyas características del fabricante podemos conseguir. Encontraremos también algunos de estos interfaces de una manera similar que para un chip I/O, tal como el PIO. De cualquier forma, no debemos adquirir ningún chip hasta estar seguros de su interconexión correcta, a menos que lo hayamos visto resuelto en alguna revista o libro. Pero aun así, se deben estudiar todo lo posible las hojas de datos técnicos publicadas por los fabricantes.

Es conveniente decir aquí que algunos datos contenidos en las hojas técnicas del ZN427 son algo confusos, como el orden exacto de las líneas de datos. En algún momento las denomina de distinta forma que en otros. Se dice, por ejemplo, que DØ corresponde al bit 8 y D7 como bit 1. En este caso concreto, si utilizamos las conexiones de la forma indicada en este libro, no nos equivocaremos.

Utilización del ADC

Para lo siguiente, suponemos que el PIO se ha conectado como se dijo anteriormente, y que el ZN427 está conectado como se indica en la Figura 4.9.

Para verificar la salida del ADC deberemos girar el potenciómetro de 1Ø K hasta llevarlo a su posición media, y teclearemos el siguiente programa:

```
10 REM *** DISPONER LADO A DEL PIO
20 REM *** PA0 SALIDA, RESTO ENTRADAS
30 OUT 95,255: OUT 95,254
40 REM *** PONER SC ALTA
50 OUT 31,1
60 REM *** PRESENTAR 40 LECTURAS DEL ADC
```

```
70  FOR I=1 TO 40
80  REM *** IMPULSO A SC
90  OUT 31,0: OUT 31,1
100 REM *** PRESENTAR VALORES EN COLUMNAS
110 PRINT IN 31,
120 NEXT I
130 REM *** BORRAR PANTALLA Y CONTINUAR
140 CLS : GO TO 30
```

Este programa simplemente lee los datos del ADC y presenta los números que encuentra en dos columnas en la pantalla por medio de un bucle continuo. Comienza poniendo el lado A del PIO, con PAØ, que está conectada a SC, como salida y el resto como entradas. Sólo utiliza 7 de los bits disponibles, que da suficiente precisión para nuestros propósitos. La línea SC se pulsa para comandar el ADC y que éste haga la conversión. Dará tiempo suficiente de realizar completamente la conversión antes de encontrar la sentencia IN de la línea 11Ø.

Podemos ver que si la luz varía en la LDR, los números mostrados mediante este programa variarán también, siendo mayores cuanta más luz incida en la LDR. Todos los números serán impares, desde el 1 al 255, pues la sentencia IN siempre leerá un 1 en el LSB (bit menos significativo) del PAØ. Esto es debido al hecho de que el 1 de la izquierda de esta línea fue puesto por la línea 9Ø y el PIO siempre devuelve el último estado que se ha puesto en una línea de salida.

Con este programa probamos el circuito, ajustando además el potenciómetro para conseguir el mejor efecto. Diferentes condiciones de iluminación requerirán diferentes posiciones del potenciómetro. Veremos que la LDR es un dispositivo muy sensible que se puede utilizar en un rango muy amplio de condiciones de iluminación.

Los números no reaccionarán instantáneamente al cambiar los niveles de luz, y leeremos el mismo valor varias veces antes del nuevo. Además, frecuentemente, el número oscilará entre dos valores próximos dependiendo del valor exacto del voltaje de entrada del ADC. Los condensadores mencionados antes absorben algunas de estas variaciones y podemos cambiar los valores de los mismos buscando mayor eficacia. En cualquier caso, es siempre mejor hacer la lectura de un rango de niveles más que números específicos, cuando se utiliza un ADC. También es conveniente leer los números varias veces y comparar una lectura con otra hasta que veamos que el ADC ha estabilizado dicho valor.

Aplicaciones

La primera aplicación para un dispositivo como éste debería ser en el ratón robot. Mediante el uso de un solo nivel umbral para la actuación de un interruptor electrónico no obtendremos resultados tan espectaculares como con el control por programa, dependiendo la actuación del valor de varios niveles de iluminación. Por ejemplo, un robot guiado por luz podría programarse para mirar primero un nivel de luz bajo y dirigirse hacia la fuente de luz detectando los sucesivos incrementos de nivel de luz. Cuando encuentra el valor máximo, el robot informaría al ordenador y giraría 180° para comenzar la búsqueda de otra fuente de luz.

Otras variables pueden ser monitorizadas del mismo modo. Por ejemplo, el estado de las baterías podría estar vigilándose continuamente, de manera que, al bajar, el robot podría buscar una fuente particular de luz, por ejemplo, coloreada con un filtro y usando una LDR especial, o mediante otro filtro. El robot se dirigiría a esta luz, con lo que, automáticamente, chocaría con un punto de alimentación desnudo que le permitiría recargar las baterías.

La lista de actividades inteligentes que puede confeccionarse es infinita. La máquina inteligente total, buscada por la robótica, aunque lejana, podemos ya imaginarla como posible, conociendo las posibilidades de control, con sensores y servos que pueden gobernarse y chequearse continuamente, que es como funciona el cuerpo humano.

El siguiente capítulo describe un tipo de robots que actúan de una manera similar a la de un brazo humano. Podremos diseñar y construir nuestros propios mecanismos, pero este objetivo sale fuera de las pretensiones de este libro, así que solamente se describirán algunos brazos de robot comerciales que pueden adquirirse normalmente.

5
Brazos robots

Introducción

La robótica es una técnica joven, y muchos de los modernos robots manejan fundamentalmente el movimiento, en general, y, particularmente, se han desarrollado mucho los llamados brazos robots que, en cierto modo, son parecidos a los brazos humanos. El ratón robot del Capítulo 3 nos proporcionó un ejemplo de sistema simple para investigar la locomoción inteligente. En este capítulo se describirán los brazos robots. Está dirigido a proporcionar una muestra de los tipos de dispositivos mecánicos baratos con los que podamos realizar el control. La única forma de que seamos totalmente capaces de apreciar el control de brazos robots es intentarlo nosotros mismos. Todos los brazos que a continuación se mencionan vienen con un manual que explica la utilización del mismo.

Otra alternativa es la utilización del interface del motor paso a paso descrito en el capítulo anterior, diseñando nuestro propio brazo mecánico y realizando el interface con el Spectrum. Esto requiere una razonable cantidad de destreza mecánica y disponer de un taller perfectamente equipado con piezas mecánicas variadas, tales como ruedas dentadas, engranajes, etc., y de una dotación de herramientas considerable. No hay problemas en la adquisición de motores paso a paso, ni de la electrónica que vayamos a necesitar. Sin embargo, si no estamos suficientemente versados en la técnica de construcciones mecánicas, deberíamos iniciar nuestra experimentación con un brazo robot adquirido en el mercado, realizando nosotros su interface con el Spectrum. Este capítulo comienza describiendo uno de los brazos robots más baratos del mercado, con un interface para el Spectrum.

A lo largo de la lectura de este capítulo iremos asimilando los principios e interfaces descritos en el libro anteriormente. Al tratarse de controlar un brazo robot, podremos pensar en añadirle en su

extremo un detector de luz o proporcionarle la facultad de hablar para informarnos del estado del sistema en cualquier momento. Mediante estos procedimientos, el sistema llegará a ser más y más un verdadero robot, emulando las funciones humanas.

Los siguientes apartados examinan algunos otros brazos robots que se encuentran en el mercado. La robótica es un campo en continua y rápida expansión, pero los brazos elegidos son de tipo general, que aseguran que los nuevos dispositivos que puedan aparecer lleguen a sernos familiares habiendo comprendido bien los presentes.

El Apéndice 4 proporciona las direcciones de las compañías mencionadas antes, que se dedican a la venta de dispositivos de este tipo, con las que podemos contactar para tener una idea de precios y disponibilidad.

Servomotores y motores paso a paso

La mayoría de los brazos robots que podemos conseguir se controlan y actúan mediante *motores paso a paso.* Los motores paso a paso son periféricos naturales de los ordenadores, pues son dispositivos digitales y proporcionan resultados repetibles con precisión. Sin embargo, los sistemas que los utilizan son caros. Por esta razón, el primer brazo que se describe en este capítulo es una máquina de bajo precio que funciona por medio de motores de corriente continua, con una realimentación interna para verificar continuamente su posición. El movimiento del brazo no es tan preciso y repetible como el de los brazos movidos y controlados por motores paso a paso que veremos más tarde, pero es uno de los métodos menos costosos de iniciarse en la robótica, y excelente para la experimentación.

El principio general del control utilizado en el brazo que vamos a describir está basado en la utilización del *servomotor,* que no es más que un motor de corriente continua, de los que se utilizan, por ejemplo, en los trenes eléctricos, pero que su salida, que es la que mueve los mecanismos del robot, dispone de un dispositivo de control que informa de su posición al propio motor. Es un motor convencional, lo que ocurre es que girará mientras esté conectada la batería de alimentación, dejando de hacerlo cuando se desconecte. Para el control robótico es necesario conocer el ángulo que ha girado el motor, conocer su posición actual para, por algún procedimiento de realimentación, desconectarlo cuando llegue a la posición en la cual deba quedarse. Hay muchos tipos de dispositivos de realimentación para esta aplicación. En la figura 5.1 se muestra uno de ellos.

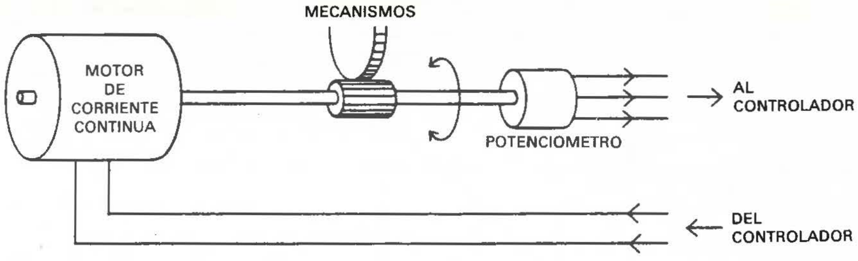


Fig. 5.1. El servomotor.

El eje del motor está dibujado rígidamente conectado a una resistencia variable o "potenciómetro". Así como el eje gira, variará la posición del cursor del potenciómetro. Si el eje pudiera girar dando varias vueltas, se debería utilizar un potenciómetro especial "multivueltas".

El controlador es el responsable de conectar el motor y monitorizar la resistencia del potenciómetro. Cuando éste alcance el valor correcto, el motor deberá ser desconectado. Esto permite al controlador saber la posición exacta de la palanca que acciona el brazo en cualquier momento. El controlador convierte efectivamente cualquier instrucción para mover la palanca hacia una posición dada, para un valor requerido de resistencia. Entonces conecta el motor en la dirección correcta hasta que el valor de resistencia se alcanza.

La forma en que funciona un controlador de este tipo está fuera del alcance de este libro, por lo que el brazo descrito hará esta función con algunos circuitos electrónicos especiales, de los cuales el usuario del ordenador no tiene noticia. Decimos que este control es "transparente" al usuario. Todo lo que el ordenador deberá hacer es enviar bytes hacia un puerto de salida, y la electrónica del brazo robot controlará el movimiento solicitado.

El brazo robot Micro Grasp

El Micro Grasp Powertran es un brazo robot completo que se vende en forma de kit, y está configurado para ser gobernado por un ordenador Sinclair ZX81. Este apartado describe el brazo, muestra cómo se puede conectar al Spectrum y apunta alguno de los controles que podemos experimentar con este pequeño y espléndido periférico.

La Figura 5.2 muestra un esquema del brazo, sujeto sobre una

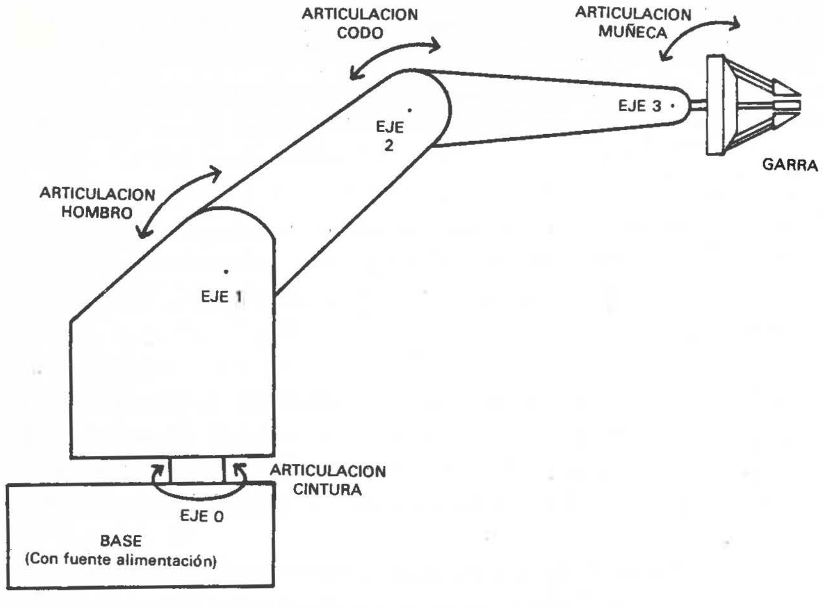


Fig. 5.2. Esquema del Micro Grasp.

base, y con una garra en el extremo que le permite coger objetos. En la base está ubicada la fuente de alimentación, que requiere una entrada de tensión de red de 220-240 V, con un consumo de unos 100 mA, la cual produce todas las tensiones requeridas por el brazo. Los componentes de la unidad de alimentación están pensados para proporcionar estabilidad, y están físicamente dispuestos de manera que se obtenga la máxima seguridad frente a la tensión de red que aparezca en cualquier parte del brazo o de su electrónica. El brazo pesa 8,6 kg, puede cargar 0,45 kg en el extremo cuando está totalmente extendido, y tiene una fuerza de agarre de 0,9 kg. La carga de cualquier brazo robot personal nunca es demasiado grande. Sólo en el área industrial se encuentran brazos que pueden llevar un peso superior a unos cuantos kilos.

La longitud o envergadura total del brazo es de 290 mm, y tiene un volumen de operación comprendido dentro de una esfera de 250 mm de radio. Tiene cuatro ejes de rotación; cada uno puede ser controlado con una precisión angular de 1 en 256. La capacidad de repetición del movimiento del brazo, medida en la garra, es de una precisión de ±6 mm. El brazo puede moverse con una velocidad de hasta 125 mm/s en su extremo.

En la base existe un motor que permite el giro de todo el brazo

alrededor de un eje vertical. Este eje se denomina articulación de garganta o articulación de cintura. También se encuentra en la base el potenciómetro de realimentación necesario para asegurar que el controlador hace girar la junta de cintura hacia la posición angular correcta comandada por el ordenador. Al eje mencionado se le llama eje Ø, y tiene aproximadamente 170° de movimiento desde un extremo hasta el otro.

El eje 1 es la articulación del hombro y tiene aproximadamente 105° de movimiento. El eje 2 es la articulación de codo, y el eje 3 es la articulación de muñeca, cada una de éstas con un movimiento aproximado de 160°. Todas estas articulaciones tienen sus motores y potenciómetros de realimentación montados en las propias juntas, con los motores funcionando a través de cajas de engranajes. Los robots que se describen en el último apartado tienen los motores montados cerca de la base y transmiten el movimiento a través de sistemas de poleas y correas de transmisión a las articulaciones que tienen que mover. Esto es lo que se hace normalmente. En este sentido, el Micro Grasp es un brazo robot atípico.

El control final es el movimiento de la garra, que puede tener tan sólo dos estados, o abierta o cerrada, sin posiciones intermedias. Esto se controla por medio de un quinto motor que hace girar un tornillo dentro de su tuerca empujando los dedos de la garra para cerrarlos, o atrayéndolos para abrirlos. Teóricamente, este motor podría valer para situar la garra en posiciones intermedias, si se le añadiera el correspondiente potenciómetro de realimentación, pero no se ha realizado de este modo en el sistema standard.

Controlando el grado y la dirección de los motores, la garra podrá tomar cualquier cosa que esté situada dentro de la esfera determinada por el radio máximo alrededor de la base. Mover la garra de un punto a otro es una acción relativamente fácil de llevar a cabo. Sin embargo, planificar el movimiento de tal modo que describa una curva predeterminada, o bien una línea recta, es un problema no tan fácil y muy interesante.

Imaginemos, por ejemplo, que el brazo está extendido, totalmente recto, horizontalmente desde la base, y deseamos mover la garra de manera que toque algún punto cercano a la base. La Figura 5.3 sugiere cómo puede ser realizado. Debemos estimar los ángulos A, B y C y sencillamente enviar las órdenes a las respectivas juntas para que se muevan esos ángulos. Sin embargo, supongamos que lo que queremos es que se alcance la posición final, pero no siguiendo cualquier camino, sino recorriendo el camino más corto, la línea recta. Para ello es necesario mover cada junta muy poco

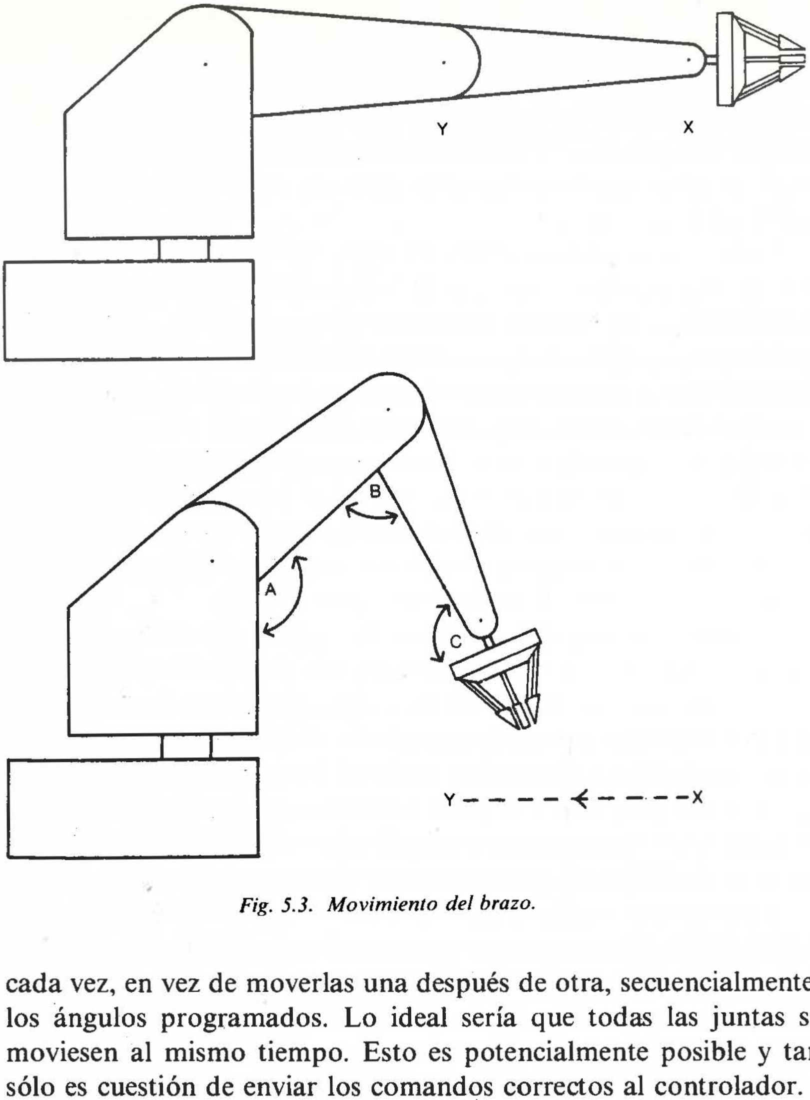


Fig. 5.3. Movimiento del brazo.

cada vez, en vez de moverlas una después de otra, secuencialmente, los ángulos programados. Lo ideal sería que todas las juntas se moviesen al mismo tiempo. Esto es potencialmente posible y tan sólo es cuestión de enviar los comandos correctos al controlador.

El control de un brazo robot es cuestión de un grado de práctica considerable. El ejercicio anterior es un ejemplo excelente para aprender a utilizar los tipos de movimientos que se necesitan para controlar los caminos recorridos por los extremos del robot en sus movimientos. Una extensión de lo anterior es intentar escribir un programa que gobierne el brazo para que escriba letras o números sobre un papel provisto de un lápiz en la garra. Conforme vayamos aprendiendo a controlar la unidad encontraremos, de una forma natural, el mejor sitio dentro del volumen que puede alcanzar la garra para colocar el papel, y las letras más sencillas que pueda

escribir. Esto podría ser el comienzo de un ambicioso proyecto para que aprenda a escribir primero su nombre, para a continuación, utilizando un sintetizador de voz, que diga las letras individuales, o todas juntas formando la palabra que escribe correctamente.

Otras tareas importantes que un brazo puede ser capaz de efectuar incluyen la toma de objetos para llevarlos de un sitio a otro. Al principio podrá hacerse sin cuidado del camino o trayectoria que siga el objeto, para después conseguir, por ejemplo, llevar un vaso de agua de un sitio a otro sin derramarla. Se trataría de tomar el vaso y llevarlo muy cerca del suelo una corta distancia, controlando el número mínimo de grados de libertad. El siguiente paso deberá ser controlar la muñeca, al mismo tiempo, para mantener el vaso hacia arriba sea cual fuese la posición que tome el resto del brazo. Comparando estos esfuerzos con la manera en que nuestro brazo realiza las cosas con tanta facilidad, nos daremos cuenta de la complejidad de dotar a las máquinas de facultades humanas, aparentemente tan sencillas de realizar. Para que el brazo realice sus movimientos de la manera más parecida a la humana, debemos fijarnos en nuestros propios movimientos reales para una acción determinada e intentar que el brazo mecánico imite todo lo posible estos movimientos.

Una vez tengamos un dispositivo robótico, las áreas experimentales que podemos ensayar son ilimitadas, y podremos inventar nuevas ideas continuamente.

Interface electrónico con el Spectrum

El brazo Micro Grasp no necesita un chip I/O de interface y, por tanto, puede ser conectado al Spectrum a través de los buses directamente. Si se utilizara un chip I/O, sus líneas de salida tendrían que emular a las líneas IORQ, WR de datos y direcciones para actuar exactamente como se describe a continuación. Esto puede hacerse, aunque requiere más interface electrónico que el método que se ha descrito, pero tiene la ventaja de que pueden utilizarse varios dispositivos robóticos al mismo tiempo.

El Interface Universal suministrado con el Micro Grasp está preparado para conectarse al ZX81, y requiere una pequeña modificación para trabajar con el Spectrum. En primer lugar, en el ZX81 el interface ocupa espacio de memoria en vez de I/O, y esto requiere línea MREQ en lugar de IORQ, además de las líneas del bus de direcciones que deben ser enviadas al interface. Toda la capacidad de direccionamiento de memoria de la CPU se utiliza en el Spectrum,

como se ha visto en el Capítulo 1, por lo que utilizaremos los puertos I/O como se ha hecho en los proyectos anteriores en este libro.

La Figura 5.4 muestra cómo deben hacerse las conexiones en el conector posterior del Spectrum. La línea MREQ de la tarjeta del Micro Grasp deberá unirse a la IORQ del Spectrum. A2, A1, AØ y WR del Micro Grasp se conectarán a las A7, A6, A5 y WR, respectivamente, del Spectrum. El bus de datos se conectará directamente como se muestra, y deberá tener una línea común de Ø V contra la cual se hace la referencia de todos los niveles lógicos.

Además de las conexiones anteriores, el Micro Grasp tiene una sección de decodificación de direcciones en su placa, que debe ser desactivada antes de efectuar las anteriores conexiones. Esto se consigue dejando libre el zócalo de IC2 (quitando el chip) y cortocircuitando el terminal 13 a los +5 V que están en el terminal 24, como se dibuja en la figura.

Cuando decidamos conectar el Micro Grasp a nuestro Spectrum, debemos emplear el conector de circuito impreso correcto para la tarjeta del controlador y soldar los hilos a él. No conviene soldar los hilos directamente a la placa, entre otras cosas porque resultará menos flexible la utilización cuando haya que desconectarlo. También, como es usual, recordemos que no deben utilizarse hilos largos desde el Spectrum hacia circuitos externos, que pueden cargar los buses e impedir que el ordenador funcione. Deberemos tomar todas las precauciones sugeridas en la construcción del interface para el PIO en el capítulo anterior, y apagar inmediatamente el ordenador y circuitos auxiliares si ocurre algo raro, así como inspeccionar el circuito con gran cuidado antes de continuar.

Gobierno del robot

Con las conexiones realizadas como se ha dicho anteriormente, los grados de libertad del robot se controlan enviando números de 8 bits a las direcciones I/O asociadas con las distintas funciones. Esos números podrán variar entre Ø y 255, y equivalen a los ángulos que pueden desplazarse cada eje. Por ejemplo, si el puerto I/O para el eje Ø recibe el número 128, el brazo entero girará hasta colocarse en la posición media de su base. Los números bajos harán que gire en un sentido y los altos en el otro. Los cuatro ejes se gobiernan de la misma manera, excepto la garra, que tiene un control más simple. Hay dos puertos I/O para la garra, uno para la orden "abrir" y el otro para la de "cerrar". Cualquier instrucción OUT enviada al

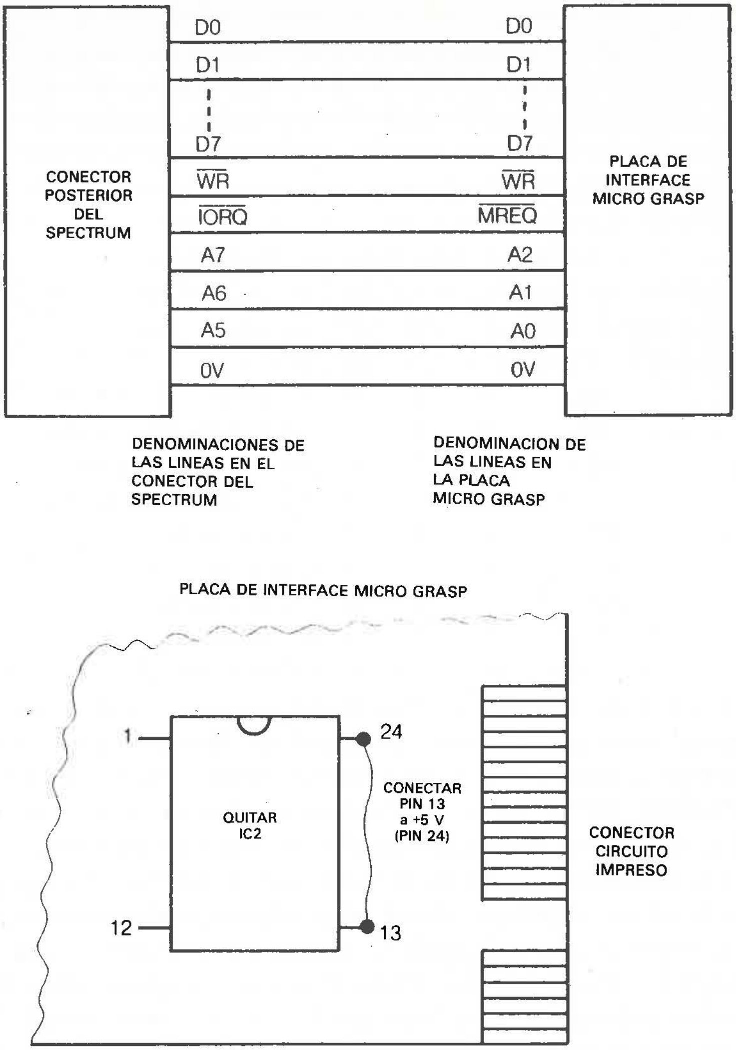


Fig. 5.4. Interface para Spectrum.

puerto de "abrir", abrirá la garra, mientras que un OUT en el otro, la cerrará. El dato preciso que se envíe es irrelevante. La electrónica de este control hay que tomarla en cuenta y está contenida en la placa de interface del Micro Grasp. Como podemos ver, se requieren seis puertos I/O, y las tres líneas del bus de direcciones que se envían al interface proporcionan ocho, dejando dos libres.

La Tabla 5.1 define las acciones de los puertos I/O en el Micro Grasp, con las conexiones de la figura 5.4. Recordemos que las

líneas AØ-A4 son todas 1. Así, si tecleamos OUT 31,Ø, la junta de muñeca del Micro Grasp girará en un sentido dando una vuelta completa. La instrucción OUT 223,Ø abrirá la garra, como también lo hará la instrucción OUT 223,45 o cualquiera dirigida al puerto 223. La placa de interface sólo utiliza el hecho de que este puerto sea contactado, no el dato real que se le envíe.

Tabla 5.1. Funciones del Micro-Grasp

Puerto I/O	*A7*	*A6*	*A5*	*Eje contactado*
31	Ø	Ø	Ø	Eje Ø junta de cintura.
63	Ø	Ø	1	Eje 1 junta de hombro.
95	Ø	1	Ø	Eje 2 junta de codo.
127	Ø	1	1	Eje 3 junta de muñeca.
159	1	Ø	Ø	No utilizado.
191	1	Ø	1	Garra cerrada.
223	1	1	Ø	Garra abierta.
255	1	1	1	No utilizado.

Como podemos ver, el gobierno del robot es sencillo y no hay ninguna tarea que el brazo pueda realizar físicamente que nadie impida programarla. Como se ha indicado anteriormente, es comparativamente fácil gobernar el brazo desde un lugar a otro, pero es difícil, en cambio, hacer que siga una determinada trayectoria.

Un interesante problema es hacer que el sistema aprenda una serie de tareas. Necesitaremos un programa que nos permita controlar el brazo a voluntad desde el teclado del ordenador. Una vez conseguido, tendremos que añadir una rutina suplementaria para recordar cada acción que hayamos hecho con anterioridad. Las operaciones podrán entonces enviarse al robot una por una, a la velocidad más adecuada para que repita la secuencia de movimientos.

Conforme vayamos experimentando con más y más funciones inteligentes, encontraremos que necesitaremos utilizar bastante memoria, con lo que el disponer de la versión del Spectrum más capaz es una gran ventaja.

El Armdroid

Un brazo robot más caro, pero más preciso que el anterior es el Colne Robotics Armdroid, que está dibujado en la Figura 5.5. Aquí

vemos que los ejes son similares a los del Micro Grasp, pero los motores están montados alrededor de la base de la máquina. Hay seis motores, todos ellos paso a paso, que utilizan el control de "bucle abierto", es decir, sin realimentación, por lo que se deberá controlar por software el número de pasos que han utilizado los motores para los distintos movimientos. Existe un conjunto de interruptores de posición inicial extra que actúan como se describió en el Capítulo 4.

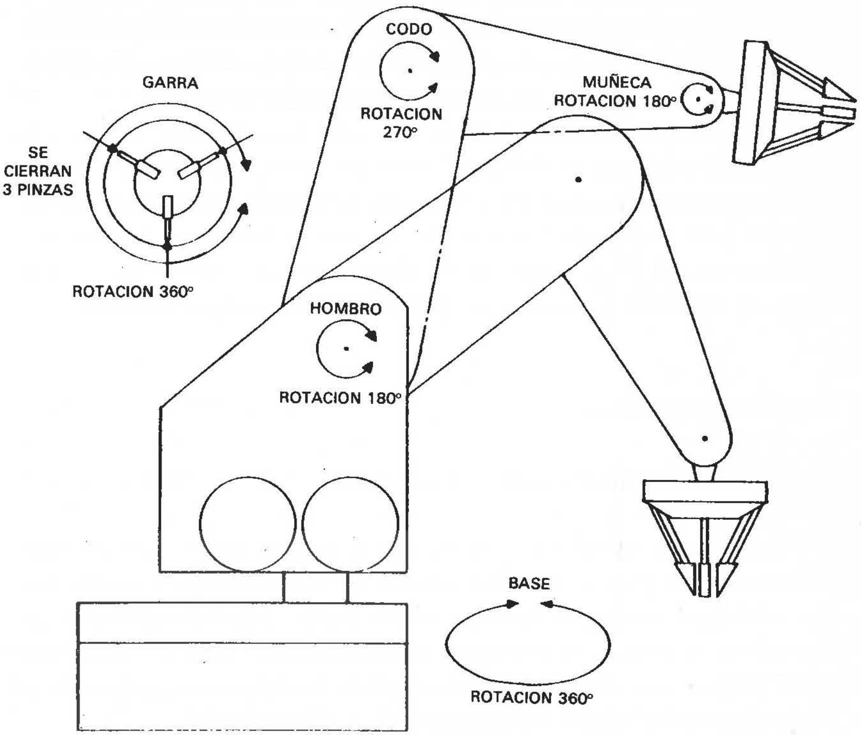


Fig. 5.5. Armdroid (reproducido por cortesía de Colne Robotics Co. Ltd.).

Como muestra la Figura 5.5, hay una gran similitud entre este robot y el que se describió antes. Es más sofisticado y preciso, pero su precio es el doble. Los cinco ángulos de movimiento están indicados en el diagrama e incluye la rotación de la garra con 360° de movimiento. Además, la garra se puede controlar continuamente por uno de los motores. Esto hace que el sistema sea muy flexible y, como los motores no están montados en el brazo, el conjunto es muy ligero, con el centro de gravedad muy bajo, con lo que se consigue mejor estabilidad. La base alberga al controlador de los motores. El paquete con la fuente de alimentación se coloca fuera.

Cada motor gobierna un árbol y una polea. Mediante una correa de transmisión se mueve otra polea colocada en la junta que debe accionarse. El engranaje es ligero pero fuerte, y debe ser ajustado para evitar que los motores paso a paso no se deslicen o ignoren pasos. Si se utilizan sensores de posición inicial mediante software, habrá que verificar los sucesos de este tipo que puedan ocurrir. Para hacerlo, el programa deberá enviar el motor a su posición inicial y verificar que el sensor se activa. El sistema deberá contar el número de pasos necesarios para corregir errores y, por consiguiente, avisar al usuario cuántos pasos erróneos se han producido por cada articulación. Esto constituye un método de autodiagnóstico.

El brazo puede alcanzar hasta 43Ø mm desde la base y tiene una precisión posicional de 4 mm. El peso del robot, sin incluir la fuente de alimentación, es de 3,5 kg, con una capacidad de carga de Ø,3 kg cuando está totalmente extendido. La fuerza de la garra es aproximadamente de 2 kg. El robot se adapta bien al control fino y a la fácil manipulación, aunque no puede manejar cargas pesadas.

Interface electrónico

Hay un interface standard para el Spectrum, con software para manejarlo.

El interface real se hace vía un puerto I/O de 8 bits. Tres de estos bits se utilizan para seleccionar uno de los seis motores, y cuatro bits para manejar las cuatro fases del motor paso a paso seleccionado. El octavo bit se utiliza para señalar la dirección del dato. Cuando este bit es bajo, el dato puede ser enviado hacia los motores y, cuando es alto, pueden ser leídos los sensores de posición inicial.

El Armdroid tiene una placa de interface para los motores paso a paso, que convierte las líneas digitales de fase de los motores en corriente para ellos. El software tiene que conmutar correctamente las fases para hacer el paso bien hecho. Esto sería una excelente aplicación del SAA1Ø27 para ahorrar software en la realización de esta tarea.

El interface para el Spectrum consiste en una placa que se conecta al conector posterior del Spectrum, y realiza el interface por medio del puerto de 8 bits. En un cassette se suministra suficiente sofware para permitirnos experimentar con el brazo, y los listados de los programas, que también se suministran, nos permitirán comprender el funcionamiento de los mismos y utilizarlos para añadir o variarlos para otras aplicaciones.

El método de interconexión, pues, del Armdroid con el Spectrum es una mera cuestión de enchufarlo y encenderlo.

Cyber 31Ø

La Figura 5.6 muestra un brazo llamado el Cyber 310, que se controla de manera similar al último ejemplo, pero que no tiene actualmente interface para el Spectrum. Se controla por medio de un puerto paralelo de 8 bits, denominado un puerto "Centronics". Este puerto contiene simplemente ocho líneas de datos, una línea de

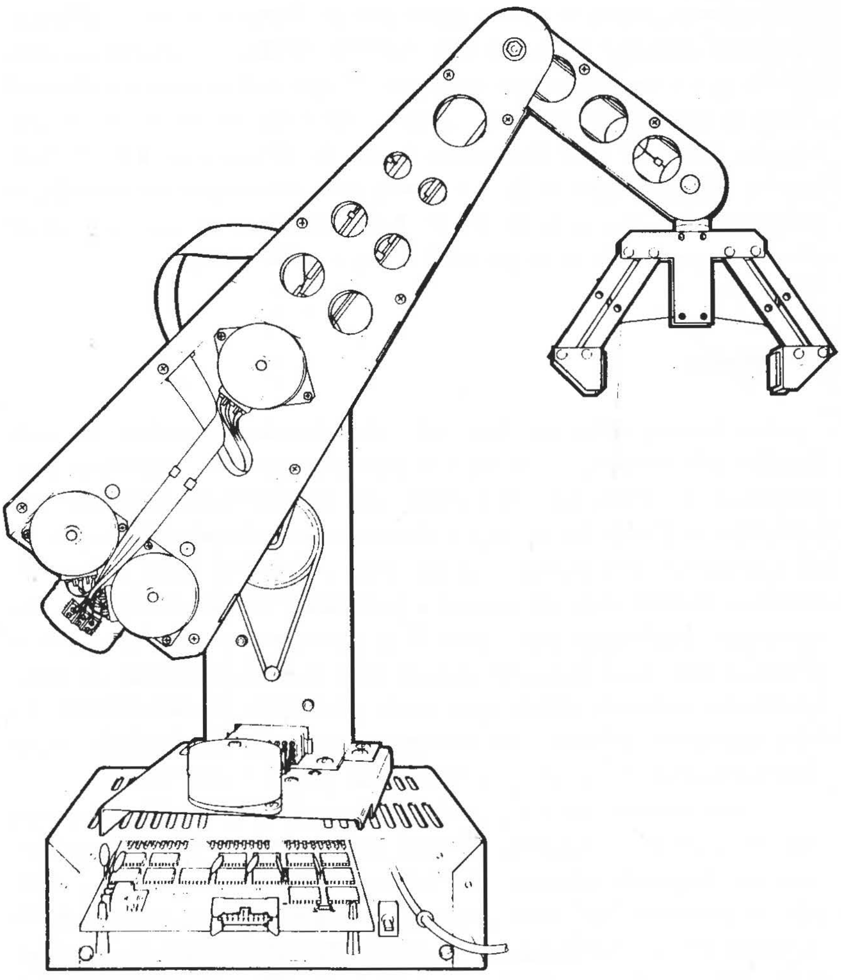


Fig. 5.6. Cyber 310 (reproducido por cortesía de Cyber Robotic Ltd.).

"ocupación" y una línea de "marcación". Los datos se envían al puerto, y su acción se realiza de manera similar a la del sintetizador del Capítulo 4. La línea INTR es similar a la de "ocupación" y la WR a la de "marcación". Por este procedimiento se seleccionan los motores y las fases a través de las líneas correspondientes. Esto requiere un software preciso y debería ser realizado en código máquina, pues el BASIC puede ser muy lento para el control de las fases. Otra solución puede ser la utilización del SAA1Ø27, como se ha sugerido en el apartado anterior, y el programa podría escribirse en BASIC.

Este brazo, como se ve en la figura, tiene algunos de los motores montados en un lateral del mismo a media altura. La mayoría de los motores transmiten el movimiento por medio de correas y cables de acero inoxidable. El brazo puede moverse de una forma muy flexible debido a su particular construcción. Tiene seis grados de libertad como el Armdroid, pero es capaz, generalmente, de moverse con ángulos mayores. La resolución de su movimiento es Ø,9-1,5 mm, que es una gran precisión, y cuando está totalmente extendido, la parte exterior está a la altura de 82Ø mm desde la base. La carga máxima que puede manejar es Ø,25 kg y pesa 15 kg.

Conclusión

Los brazos descritos han sido elegidos para mostrar los más baratos del mercado. Los precios específicos no se mencionan aquí debido a las continuas revisiones que los fabricantes hacen. Sin embargo, se puede hacer una estimación aproximada de los precios en Inglaterra, que pueden oscilar entre 25Ø y 75Ø libras esterlinas. Mejores prestaciones, en cuanto a precisión y flexibilidad, se pueden conseguir con brazos más caros. Hay varios brazos en el mercado, y el mejor plan para comprar uno de ellos es revisar revistas de ordenadores y robótica donde aparezcan ofertas de los fabricantes. En ellas aparecen, además, las novedades que van apareciendo continuamente con revisiones y advertencias para su utilización.

Como hemos mencionado previamente, la alternativa de todo esto es diseñar y construir nuestro propio brazo. No es imposible hacerlo. Se puede empezar con un dispositivo de un solo eje y obtener experiencia suficiente para continuar. Deberá proveerse de material en una buena tienda de modelismo para elegir los mecanismos, tales como engranajes y poleas, láminas de aluminio para revestir los laterales del brazo. Si es posible, conviene estudiar los

mecanismos de algún brazo comercial para empezar a trabajar sobre una idea ya implementada en ellos.

Uno de los principales inconvenientes de la mayoría de los brazos comerciales es que no proporcionan apenas nada en el campo de los sensores para realimentación. Este libro ha sugerido algunos que se pueden considerar para añadir a los brazos robots, con el fin de hacerlos más inteligentes.

Es conveniente realizar experiencias con muchos sensores robóticos y controles de los mismos. Con ello se pretende realizar máquinas lo más parecidas posible a las personas, con más cualidades cada vez, con más inteligencia.

Apéndice
1
Códigos binarios, hexadecimal y ASCII

Introducción

En el manual del Spectrum se introducen algunos de los conceptos que vamos a contemplar en este Apéndice, pero sólo desde el punto de vista de programación. En este Apéndice se tratan estas ideas con objeto de que entendemos la electrónica de las máquinas, sobre todo la manera en que el ordenador almacena y, por tanto, comunica los datos electrónicamente. Bajo este punto de vista es como se abordan aquí estos conceptos.

En conveniente familiarizarse con el manual del Spectrum en su totalidad antes de leer este Apéndice.

Almacenamiento de números en el ordenador

Después de haber manejado suficientemente el Spectrum y de conocer sus posibilidades, estamos en condiciones de entender el lenguaje interno del ordenador necesario para su programación. Por ejemplo, entendemos los números binarios. Veremos a continuación cómo utiliza el ordenador este tipo de numeración y por qué. Primero veremos esto considerando un sencillo dispositivo de memoria.

El dispositivo de memoria más simple que puede ilustrarnos es un interruptor eléctrico manual. Tiene dos estados, "cerrado" y "abierto", no tiene más. Cuando el interruptor está "cerrado" este estado es recordado a menos que haya alguien que lo abra. Pues bien, ésta es la esencia de todas las memorias de ordenador, aunque los conceptos involucrados en su manipulación pueden llegar a ser extraordinariamente complejos. Pero recordando este ejemplo de los interruptores, puede entenderse con facilidad cualquier dispositivo de memoria real.

Supongamos ahora que preparamos un conjunto de interrupto-

res manuales para manejar otras tantas lámparas. ¿Cuántos estados diferentes puede adoptar este cuadro de luces? Esto dependerá, desde luego, del número de interruptores. Si hubiese uno solo, tendremos dos estados, cerrado y abierto (encendido y apagado). Si fuesen dos los interruptores, podríamos tener cuatro estados diferentes de las lámparas, ambas encendidas, ambas apagadas, y las dos maneras de tener una encendida y la otra apagada. No es coincidencia que el número total de estados (4) sea el cuadrado de dos, esto es 4 = 2*2. En efecto, puede generalizarse este cálculo para otras cantidades. Si hay tres luces, el número de estados sería 2*2*2 = 8 (2 elevado a 3), y así sucesivamente. Por este procedimiento vemos que si hay 8 luces, habrá 2 elevado a 8 (= 256) combinaciones diferentes.

Este razonamiento basado en las potencias de dos, dependiendo del número de "interruptores", es fundamental en los ordenadores digitales.

Almacenamiento de caracteres

Como sabemos, una de las cualidades principales de un ordenador es que posee memoria. No es suficiente ser un calculador muy eficiente, cosa difícil sin memorias, sino que es esencial, para todas las tareas que se le encomiendan a un ordenador, que disponga de mucha cantidad de memoria para almacenar datos, variables, letras, números, textos de cualquier tipo, etc. Consideremos, por un instante, cómo podría implementarse esta memoria. Podríamos, quizás, producir dispositivos electrónicos, cada uno con diferentes estados para ser capaz de almacenar cualquier número, letra, etc. Cuando deseáramos almacenar una carta de un amigo tendríamos que almacenar cada carácter en su propio dispositivo, para poder leer en el futuro el estado de cada uno y reproducir la carta.

Por unas cuantas razones, la decisión de hacer dispositivos electrónicos con sólo dos estados ha sido tomada hace muchos años. La manera de almacenar caracteres o números en estos dispositivos, que no son más que agrupaciones de "interruptores", se basa en adjudicar a cada combinación de posibles estados de cada grupo de interruptores un carácter o letra o número, etc., mediante un determinado código. El código más comúnmente empleado internacionalmente es el ASCII (American Standard Code for Information Interchange). Este código está basado en un grupo de siete "interruptores", cada uno de ellos con dos estados, cerrados o abiertos. Esto proporciona 128 combinaciones diferentes (2 elevado a 7), cada una

de ellas asimilada a un símbolo. En el Spectrum, esta lista de caracteres está extendida al doble, lo que explica la riqueza y variedad de símbolos distintos que se pueden generar en la pantalla con el Spectrum; algunos de los cuales los podemos crear nosotros mismos. En el manual del Spectrum se relaciona la lista completa de los caracteres con sus códigos y algunos programas útiles para emplearlos. Para representar cualquiera de esos 256 caracteres diferentes necesitaremos grupos de 8 bits, como ya sabíamos cuando programábamos el Spectrum.

Imaginemos un banco de memoria compuesto de muchas filas de interruptores, cada una de ellas con 8 de ellos. Cada una de estas filas podrá representar, gracias al código de carácter, uno de los 256 posibles que una persona puede reconocer en un trozo de texto, tales como letras, números, gráficos, etc. Si dispusiéramos de suficientes filas, se podrían colocar los interruptores de forma que podríamos "escribir" con ellos las obras de Cervantes, por ejemplo, carácter a carácter, en este banco de memoria. No hay otro límite que el práctico en este tipo de memoria.

Cálculos y bases de numeración

Hay otras cosas que nos gustaría almacenar en este banco, cálculos por ejemplo. Sería una gran restricción para un ordenador que no fuera capaz de efectuar cálculos. Esto implica que además de poder almacenar números que significan códigos de caracteres —los dígitos del Ø al 9 son símbolos o caracteres del código ASCII— se pudieran almacenar números como tales, y poder realizar cálculos con ellos. Para entender la diferencia, imaginemos la realización de cálculos con números romanos. El sistema romano no es más que un código, como el ASCII, de representar números, pero realizar una larga división con ellos es poco menos que imposible.

Los números, pues, deben tratarse y almacenarse de una manera tal que facilite los posteriores cálculos con ellos, encontrando alguna norma o "base" de representación idónea. Uno de los condicionamientos que nos impide adentrarnos con facilidad en el mundo de los ordenadores es nuestra costumbre de contar los números grandes de diez en diez, o tomando como base el número 1Ø, como consecuencia, probablemente, de tener esa cantidad de dedos en las manos. El sistema de numeración basado en el número 1Ø o de "base 1Ø", se denomina también "sistema denario". Empezamos a contar a partir del cero hasta el 9. Cuando necesitamos números mayores,

añadimos otro número a la izquierda, con lo que podemos alcanzar hasta el 99. Creando otro nuevo dígito a la izquierda de los anteriores, llegaríamos hasta el 999, a partir del 1ØØ (1Ø al cuadrado). Así llegamos hasta el 1ØØØ (1Ø al cubo, o lo que es lo mismo, 1Ø multiplicado por sí mismo tres veces), y así sucesivamente. Es decir, con nuestro sistema de numeración en base 1Ø, y con tan sólo 1Ø dígitos (del Ø al 9), podemos representar cualquier número por grande que sea. Esto que ocurre con el sistema de base 1Ø, ocurre también con cualquier sistema de numeración que hubiésemos elegido. Si utilizáramos tan sólo dos dígitos (el Ø y el 1), podríamos contar de la misma manera. Este sistema se denomina de "base 2" o "binario". Aunque ya conocíamos este sistema de numeración, reproducimos a continuación una tabla que aparece en el manual del Spectrum para refrescar las ideas y recordar el procedimiento de formar números utilizando este sistema de numeración y con el sistema hexadecimal, que veremos más tarde.

Obsérvese que cada vez que se necesita una potencia de dos (por ejemplo 2, 4, 8, 16...) se ha de añadir un dígito más, del mismo modo que se necesitaba en el sistema de base 1Ø cuando aparecían potencias de 1Ø. Estudiemos esta tabla y tratemos de ampliarla como ejercicio:

Decimal	*Binario*	*Hexadecimal*
Ø	Ø	Ø
1	1	1
2	1Ø	2
3	11	3
4	1ØØ	4
5	1Ø1	5
6	11Ø	6
7	111	7
8	1ØØØ	8
9	1ØØ1	9
1Ø	1Ø1Ø	A
11	1Ø11	B
12	11ØØ	C
13	11Ø1	D
14	111Ø	E
15	1111	F
16	1ØØØØ	1Ø

Binario

Para llevar al mismo tiempo juntas las ideas de memoria con el sistema de numeración binario, imaginemos los interruptores con sus dos estados cerrados y abiertos, correspondientes a unos y ceros, respectivamente. Para representar el número 15, por ejemplo, el banco de memoria de interruptores necesitará cuatro de ellos, todos ellos en estado cerrado (ver tabla anterior). Los números 1 y Ø se llaman dígitos *binarios* (binary digits), o abreviadamente bits. Diremos que la tabla anterior utilizada un máximo de 5 bits.

Un método de conversión de binario en decimal lo tenemos por medio de la función BIN. Desgraciadamente, la función BIN puede actuar sobre un conjunto de unos y ceros, pero no se puede, por ejemplo, usar INPUT para introducir un número binario en la variable A$, y luego usar BIN VAL A$ para encontrar el decimal equivalente. Si queremos encontrar el valor decimal del número binario 1ØØ1Ø11, por ejemplo, teclearemos

```
PRINT BIN 1ØØ1Ø11
```

a lo que el ordenador constestará con: 75.

El tamaño de los números determina cuántos bits se necesitan para representarlos, como se puede deducir de la tabla anterior. Para ser capaz de representar todos los números, sean grandes o pequeños, y organizar adecuadamente las direcciones de la memoria, los dispositivos de memoria están divididos normalmente en grupos iguales de bits para todas las localizaciones. En el Spectrum hay cabida para 8 bits en todos los compartimentos de memoria.

Esto tiene implicaciones muy importantes en cuanto a la electrónica del ordenador. Cada compartimento tiene su propia dirección electrónica y se pueden almacenar y recabar datos a voluntad, como se explicó en el Capítulo 1. Por el momento, pensemos que cada compartimento es como si fuese una de las filas de 8 interruptores que vimos anteriormente. Cada compartimento de 8 bits puede almacenar una de las 256 combinaciones de unos y ceros, es decir, está claro que puede almacenarse cualquier número desde el Ø al 255, simplemente convirtiéndolo en binario. Pero ¿cómo podríamos almacenar un número tal como el 328 o el 643 en binario? La respuesta a esta cuestión sale fuera de las pretensiones de este libro, pero bastará decir que podemos reservar varios de estos compartimentos para almacenar cualquier tipo de números mayores de 255, hasta cierto límite. Del mismo modo, utilizando varios de estos

compartimentos, y agrupándolos convenientemente, se podrán almacenar números decimales también.

Binario y electrónica

El sistema de numeración binario es un concepto muy importante para comprender la electrónica de la máquina, pues las líneas electrónicas del sistema de ordenador transportan la información de unos y ceros, en forma de niveles de tensión de + 5 V y Ø V, en todo instante. Así, el estado de las líneas de bus (ver Capítulo 1) en un instante cualquiera, se describe por un número binario como la mejor forma de representación. El bus de direcciones, por ejemplo, tiene 16 líneas; esto es, cualquier dirección puede ser representada por un número de 16 bits. Sin embargo, tanto la escritura de un número de 16 bits, como su lectura es una tarea penosa y sujeta a frecuentes errores para cualquier persona. Por ello debemos buscar un método para, aun simplificando la escritura de números binarios grandes y, por tanto, disminuyendo la posibilidad de errores y la incomodidad, seguir disponiendo de la facilidad que el sistema binario proporciona para la interpretación rápida del estado de un grupo de líneas electrónicas.

Hexadecimal y direcciones

El método buscado pudiera ser, y de hecho así se ha adoptado, el dividir los números binarios grandes en grupos de cuatro bits, y escribir cada uno de estos conjuntos de forma compacta. En el manual del Spectrum se explica cómo se hace. La tabla anterior mostraba la denominación hexadecimal que podemos aplicar a cada grupo de 4 bits.

El Capítulo 1 muestra que las direcciones que la CPU está contactando en cualquier momento aparecen como los distintos estados binarios de las 16 líneas del bus de direcciones. El mapa de direcciones del Z8Ø comienza por Ø (es decir, todas las líneas estarían a Ø V), y llega a la dirección 65.535, que corresponde con el número binario 1111 1111 1111 1111, o bien con el hexadecimal FFFF, como se puede deducir de la tabla citada. Una cuestión que se nos podría ocurrir es encontrar la dirección central de este mapa. Para ello dividiremos por dos el número total de direcciones posibles, es decir la dirección central será la 65.536/2 = 32.768 expresada en su valor

decimal. El número binario equivalente es el 1ØØØ ØØØØ ØØØØ ØØØØ, o lo que es lo mismo, el hexadecimal 8ØØØ. La representación hexadecimal de números binarios de grandes dimensiones, divididos en grupos de 4 bits, es, pues, un sistema que reúne las dos condiciones de facilidad de manejo y rapidez de conversión al sistema binario, necesario para aplicaciones puramente electrónicas. Podríamos haber empleado la conversión directa entre binario y decimal, puesto que éste último nos es tan familiar y perfectamente manipulable en BASIC, pero dicha conversión es mucho más lenta cuando se está trabajando con la electrónica del ordenador. Veámoslo con un ejemplo. Supongamos que en un momento dado queremos ver mediante la ayuda de un medidor de tensión el estado del bus de direcciones, que debería ser, por ejemplo, el equivalente en binario del número 45.741 decimal. Primero tendríamos que convertir el número decimal en binario, lo que nos tomaría algún tiempo. El resultado sería:

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	AØ
1	Ø	1	1	Ø	Ø	1	Ø	1	Ø	1	Ø	1	1	Ø	Ø

Si, en cambio, hubiéramos partido del valor hexadecimal B2AC, hubiéramos generado el número binario con gran facilidad, bien utilizando la tabla de conversión o, después de un poco de práctica, automáticamente de memoria. Por este motivo se utiliza para aplicaciones electrónicas y por cuestiones de otra índole, la notación hexadecimal como la forma de expresión simplificada del sistema binario más adecuada y fácil de asimilar. Así vemos que ya en el Capítulo 1 las direcciones del mapa de direcciones están expresadas tanto en decimal como en hexadecimal. Por supuesto que no podemos abandonar el sistema decimal, el cual, por otro lado, es el sistema de numeración único en la programación BASIC para acceder o escribir direcciones, con instrucciones tales como PEEK, POKE, IN y OUT.

Conversión decimal-hexadecimal

El siguiente programa convierte números decimales en hexadecimales y viceversa. Si estamos interesados en el método de conversión podremos extraerlo de la explicación que a continuación se describe de este programa. Si no es así, podemos teclear simplemente dicho programa para utilizarlo sin más, siempre que queramos realizar conversiones de este tipo.

```
  10 INPUT "Entrada HEX o DEC?
     (H o D)=",A$
  20 IF A$="H" THEN  GO TO 1000
  30 IF A$<>"D" THEN  GO TO 10
  40 REM Ahora conversion DEC a HEX
  50 INPUT "Introduzca no.=65535",N
  60 IF N=65535 THEN  LET H$="FFFF":
     GO TO 2000
  70 LET H$=""
  80 LET K=INT (N/4096)
  90 IF K<>0 THEN  GO SUB 500: LET H$=S$
 100 LET L=N-K*4096
 105 REM Cuantos 256's caben
 110 LET K=INT (L/256)
 120 IF K=0 THEN  IF H$<>""
     THEN  LET H$=H$+STR$ 0
 130 IF K<>0 THEN  GO SUB 500: LET H$=H$+S$
 140 LET L=L-K*256
 150 REM Cuantos 16's caben
 160 LET K=INT (L/16)
 170 IF K=0 THEN  IF H$<>"" THEN
     LET H$=H$+STR$ 0
 180 IF K<>0 THEN  GO SUB 500: LET H$=H$+S$
 190 LET L=L-K*16
 200 IF L=0 THEN  LET H$=H$+STR$ 0: GO TO 2000
 210 LET K=L: GO SUB 500
 220 LET H$=H$+S$
 230 GO TO 2000
 500 REM Rutina para cambiar digitos DEC en HEX
 510 IF K<=9 THEN  LET S$=STR$ K: RETURN
 520 LET R$="ABCDEF"
 530 LET T=K-9
 540 LET S$=R$(T TO T)
 550 RETURN
1000 REM Ahora conversion HEX a DEC
1010 INPUT "Introducir 4 digitos HEX o menos",A$
1020 LET K=0
1030 LET T=LEN A$
1040 FOR I=T TO 1 STEP -1
1050 LET K$=A$(I TO I)
1060 GO SUB 1500
1070 LET K=K+P*16^(T-I)
1080 NEXT I
1090 GO TO 3000
1500 REM Rutina para cambiar un digito HEX a DEC
```

```
1510 IF K$>="0" THEN  IF K$<="9" THEN
     LET P=VAL K$: RETURN
1520 LET P=(CODE K$)-55
1530 RETURN
2000 PRINT "El no.HEX=",H$
2010 GO TO 4000
3000 PRINT "El no.DEC=",K
4000 INPUT "Otro no.?(Y o N)",A$
4010 IF A$="Y" THEN GO TO 10
4020 CLS : PRINT "Fin del programa"
```

Para ver cómo funciona el programa, consideremos cómo se forman los números hexadecimales. Son simplemente números contados de acuerdo con la base de numeración 16, de la misma forma que contamos con base 1Ø. Los números del Ø al F pueden convertirse utilizando la tabla de este Apéndice. Los números más altos se forman de una manera similar a los del sistema decimal, como se explicó anteriormente. Para contar por encima del F en el sistema hexadecimal, tendremos que añadir un dígito a la izquierda, como cuando se cuenta por encima de 9 en decimal. Cada vez que encontremos una potencia de 16 (en hex) comenzaremos con otro dígito a la izquierda. Así:

1 (dec) = 1 (hex)
16 (dec) = 1Ø (hex)
256 (dec) = 1ØØ (hex)
4Ø96 (dec) = 1ØØØ (hex)

Esto es todo lo que necesitamos, ya que en este libro no vamos a utilizar más de 4 dígitos hexadecimales, aunque el sistema de numeración, al igual que el decimal, es válido para infinitos dígitos.

Para convertir un número hexadecimal en decimal, consideramos el siguiente ejemplo:

A35E = A*4Ø96 = 3*256 + 5*16 + E =
=1Ø*4Ø96 + 3*256 + 5*16 + 14 = 41.822 (decimal)

El último dígito de la derecha (llamado el *dígito menos significativo,* o LSD) es el número que representa las unidades. El siguiente dígito de la izquierda es el número de veces que está contenido el 16, a continuación el número de veces que está el 2565, y por último, el 40965, que es el *dígito más significativo,* o MSD. En esto se basa el programa anterior que realiza la conversión de hexadecimal.

Para realizar la conversión en el otro sentido, el programa calcula cuántas veces está contenido en el número el 4Ø965, por simple división. El número resultante sería el MSD, si no es cero. Entonces, el resto se divide por 256, que dará el dígito siguiente. El resto se vuelve a dividir, esta vez por 16, para encontrar el nuevo dígito, y el resto final será el LSD. En este libro sólo utilizamos números entre Ø y FFFF, en hexadecimal, que equivalen al Ø-65.535 en decimal. En total, 65.536 números, incluido el Ø.

El programa incluye un método de conversión de hexadecimal a decimal. Deberemos tratar de entender cómo lo hace.

Binario a decimal y viceversa

Para convertir números binarios en decimales utilizaremos un procedimiento que será fácil de entender con un ejemplo. Supongamos que necesitamos convertir el número binario 1ØØ1Ø111 a su equivalente decimal. Primeramente lo dividiríamos en grupos de 4 bits y lo pasaríamos a hexadecimal para, posteriormente, convertirlo a decimal. Pero también podríamos utilizar el camino alternativo de emplear el siguiente tipo de tabla:

128	64	32	16	8	4	2	1
1	Ø	Ø	1	Ø	1	1	1

Cuando haya un 1 tendremos que añadir el número de encima. El total de las sumas será el número buscado. En este ejemplo, el número es el 128 + 16 + 4 + 2 + 1 = 151. Para convertir números decimales en binarios se tendrán que realizar divisiones, del mismo modo que en el caso de la conversión decimal en hexadecimal. Supongamos que el número 4271 debe ser convertido en binario. El método será el siguiente:

4271/2 =	2135 resto 1 LSB
2135/2 =	1Ø67 resto 1
1Ø67/2 =	533 resto 1
533/2 =	266 resto 1
266/2 =	133 resto Ø
133/2 =	66 resto 1
66/2 =	33 resto Ø
33/2 =	16 resto 1
16/2 =	8 resto Ø

8/2 = 4 resto ∅
4/2 = 2 resto ∅
2/2 = 1 resto ∅
1/2 = ∅ resto 1MSB

(todas las siguientes divisiones darían ∅ con resto ∅).

El número binario está compuesto de los restos, con el último bit de arriba como el más significativo (MSB). Así:

4271 = 1∅∅∅∅1∅1∅1111

que es una representación de 13 bits del 4271. Si necesitamos una representación de 16 bits, simplemente añadiríamos ceros a la izquierda hasta completar los 16

4271 = ∅∅∅1 ∅∅∅∅ 1∅1∅ 1111

Utilizando la tabla, veríamos que este número equivale al número hexadecimal 1∅6F.

Notación K y M

Nuestro Spectrum tendrá 16 K ó 48 K. Esto se refiere al número de bytes que pueda almacenar la máquina en la memoria RAM (ver Capítulo 1). Como todas las direcciones de la memoria utilizan números binarios, no es natural hablar en unidades de miles de bytes, pues esto es un número de base 1∅. La potencia de 2 más cercana a un millar es 2 elevado a 1∅, o sea 1.∅24. A este número es al que le llamamos K o unidad K, en el lenguaje informático. Así pues, 16 K significa 16*1∅24, es decir 16.384. De manera similar, 48 K es 48*1∅24 = 49.152. La mayor localización de memoria en el Spectrum es la correspondiente a 65.535. Así, incluyendo la localización ∅, hay

64*1.∅24 = 65.536 = 64 Kbytes

localizaciones posibles en la máquina.

La unidad K es denominada generalmente "Kilo", como si fuese equivalente a mil, pero debemos tener en la mente que lo que en realidad significa es 1.∅24, si queremos ser precisos.

Otro término utilizado frecuentemente es Megabytes, que equi-

vale a 1.ø48.576 bytes, que es también una potencia de dos (2 elevado a 2ø). Como podemos ver, cuando dispongamos de 1 Mbytes de almacenamiento, tendremos en realidad, aproximadamente, unos 48 K más de memoria, que si tuviésemos un millón de bytes.

Apéndice 2 Electrónica, componentes y soldadura

Introducción

La electrónica digital, que incluye la mayoría de los ordenadores es, en buena medida, una cuestión de acoplar circuitos integrados convencionales, que realizan para nosotros una gran cantidad de funciones. Verdaderamente poca electrónica vamos a necesitar en este libro, y la poca que se necesita está descrita de acuerdo con las necesidades de un principiante. Este apéndice proporciona algunas ayudas a los que se inicien, las herramientas necesarias y consejos sobre la realización de soldaduras y reconocimientos de diferentes componentes.

Electricidad y electrónica

Cuando los terminales de una batería se cortocircuitan, se establece una corriente o flujo de partículas (electrones) desde un terminal hacia el otro. Puede ser útil recordar que este flujo de partículas se comporta de una manera muy parecida al agua. Por ejemplo, si tenemos una tubería muy fina, ésta no será muy buena para transportar el agua, o lo hará de manera que ésta pierda presión. Hay como una oposición relativa al paso de la corriente de agua. Decimos que en este caso, como en el caso de un material conductor de la electricidad, hay una resistencia al paso de la corriente. La pérdida de energía en dicha resistencia se puede utilizar normalmente transformándola en luz o calor.

La corriente que fluye, ocasiona normalmente un incremento de la temperatura en algunos lugares. Cuando el Spectrum se calienta, sobre todo en la versión de 48 K nos está indicando que la corriente está pasando por él. El Spectrum, desde luego, no utiliza una batería; utiliza el suministro de la red eléctrica para funcionar. La

corriente producida está haciendo funcionar al Spectrum todo el tiempo y esta cantidad de trabajo por el tiempo que está haciéndolo es lo que llamamos *potencia*. Esto es lo que ocurre cuando decimos que la potencia de un dispositivo eléctrico es, digamos, de 1 kilovatio. Es una medida de la cantidad de trabajo (o energía) que se consume cada segundo y, por consiguiente, el costo. En la microelectrónica, intentamos mantener la potencia real en lo mínimo posible, para utilizar fuentes de potencia (o fuente de alimentación) pequeñas y generar la menor cantidad de calor que podamos.

La corriente fluye por un conductor solamente si existe voltaje (diferencia de potencial eléctrico) en sus extremos, lo que puede conseguirse de varias maneras. Hay un voltaje entre los terminales de una batería, por ejemplo, conseguido por medios químicos. Hay voltajes en nuestros hogares, gracias a la red de transporte de energía. Este voltaje está continuamente variando de polaridad, por lo que se denomina tensión alterna (impropiamente, pero más familiar, corriente alterna). Las baterías, en cambio, proporcionan tensiones continuas (corriente continua) sin variaciones, ni de polaridad ni de magnitud.

La fuente de alimentación ZX proporciona una tensión continua que no es totalmente plana (constante en el tiempo), pero que nunca cambia la polaridad, es decir, puede variar su valor dependiendo de la corriente que suministre o, muy rápidamente, alrededor de un valor dado (rizado), con variaciones muy pequeñas comparadas con el valor medio de la tensión que proporciona. A este tipo de fuentes de alimentación se les denomina fuentes de alimentación sin estabilizar. Para estabilizar la tensión se utilizan circuitos adicionales, como tiene el Spectrum en su interior (figura 1.2). Un ejemplo de lo que hace un circuito de este tipo lo encontraremos en el Capítulo 2 para alimentar un proyecto allí descrito y otros que se describen posteriormente.

Con objeto de sacar el máximo partido posible de la información que existe en los buses de un ordenador, con el mínimo de potencia, los diseñadores intentan utilizar pequeñas tensiones con pequeñas corrientes. El voltaje de los circuitos lógicos es de +5 V, y, aunque a lo largo de cada línea hay un nivel de corriente detectable, normalmente no se tiene en cuenta. La detección de la información electrónica se realiza cuando lo que cambia en la línea es la tensión.

La información aparece en las líneas por la acción de interruptores electrónicos muy rápidos, en nada comparable a la velocidad de un grifo para agua. Se cierran y abren para cambiar el voltaje en la línea de acuerdo con un conjunto de secuencias, y los receptores de

estos cambios utilizan estos voltajes para cerrar y abrir asimismo sus interruptores internos. En efecto, el cableado interno de un ordenador está continuamente transportando cambios de voltajes millones de veces por segundo. Es imposible atrapar ninguna de estas transacciones si no es utilizando medios electrónicos, sincronizados con el proceso.

Para construir el hardware necesario, se requiere una cierta familiaridad con la construcción de circuitos electrónicos, y esto es lo que se comenta a continuación.

Herramientas

Las herramientas que se necesitan son simples y normalmente baratas. Se necesita un par de buenos alicates de corte para cortar el hilo de conexión y los terminales de los componentes, y deben elegirse aquellos que están construidos para este uso. No deben ser excesivamente grandes. También se necesita un destornillador de pala plana pequeño y otro grande. Será de mucha utilidad una pareja de alicates, para uso de pequeños trabajos electrónicos, así como un par de pelahilos. Es fundamental un soldador de unos 15 W con punta entre fina y mediana. y el correspondiente carrete de estaño. Puede ser conveniente tener dos soldadores, uno grande y otro pequeño. El uso de una lupa de unos 5 ó 10 aumentos puede ser de mucha ayuda para inspeccionar soldaduras. Otra importante pieza del equipamiento es un multímetro con escala de resistencias. Se usa para medir voltajes, corrientes y verificar continuidad. En el manual de instrucciones de cada multímetro se indican los procedimientos de medida en cada escala.

Componentes generales

Los componentes electrónicos más simples son: conectores, zócalos, enchufes, clavijas, lamparitas, resistencias y condensadores seguidos por: diodos, transitores y, por último, los circuitos integrados. Estos componentes se describen a continuación. Todos estos componentes se venden en tiendas especializadas, y son fáciles de encontrar.

Para expandir nuestro Spectrum, necesitamos adquirir el conector adecuado al cual hay que soldarle unos cuantos hilos. Este conector es tan popular que sólo necesitaremos pedirlo por este nombre: el

conector de expansión para el Spectrum. No obstante se indican a continuación sus características. Se trata de un conector para placa de circuito impreso de Ø,1 pulgadas de paso con doble conexión para 28 terminales, con terminales internos preparados para soldar, o para wrapinar, con los extremos abiertos, y la ranura de posicionamiento en la quinta pista. En el Apéndice 3 se describe gráficamente la disposición y denominación de todos los terminales de este conector. Los hilos soldados al conector se llevarán por el otro extremo a la placa del dispositivo que queremos conectar al Spectrum.

Cualquiera de las lamparitas utilizadas en los proyectos serán generalmente LEDS (Light Emitting Diodes), diodos emisores de luz, aunque también podrían controlarse luces más potentes. El cableado que se emplea normalmente suele ser hilo fino de cobre esmaltado con aislante plástico de diversos colores, agrupados en mazos, o individuales. Cuando se trata de realizar un cableado muy complejo en una tarjeta, se suele usar la técnica de wrapinado, que no es más que una manera de enrollar un hilo especial a los terminales adecuados a este fin. Requiere un poco de práctica, pero los montajes quedan muy limpios y las conexiones fáciles de deshacer.

El tipo de interruptores utilizados suelen ser del tipo subminiatura, y pueden ser interruptores propiamente dichos, con dos estados estables o pulsadores. Las dimensiones no deben ser muy grandes, aunque el funcionamiento eléctrico es el mismo si se utiliza uno muy capaz en lugar de otro pequeño.

Hay muchos sistemas preparados específicamente para realizar prototipos para el Spectrum en el mercado, y muchos de ellos excelentes. Con uno de estos sistemas, el trabajo de cableado se reduce considerablemente. Algunos de ellos son caros y no añaden mucha más facilidad que la utilización de placas de pistas impresas standard, con taladros para el paso de los terminales de los componentes. Un ejemplo de este tipo de placas es la placa Veroboard, a la que nos hemos referido a todo lo largo de este libro. Todas ellas tienen pistas de cobre, que pueden disponerse para las conexiones entre los componentes realizando cortes en los sitios adecuados (ver figura A.1). Los componentes se introducen desde el lado aislado hacia el lado del cobre a través de los taladros y se sueldan por este mismo lado. Los taladros están separados Ø,1 pulgadas, que es una dimensión normalizada a la que se adaptan los componentes más comunes.

La Figura A.1 muestra los componentes electrónicos más comunes con los símbolos que se emplean en los diagramas.

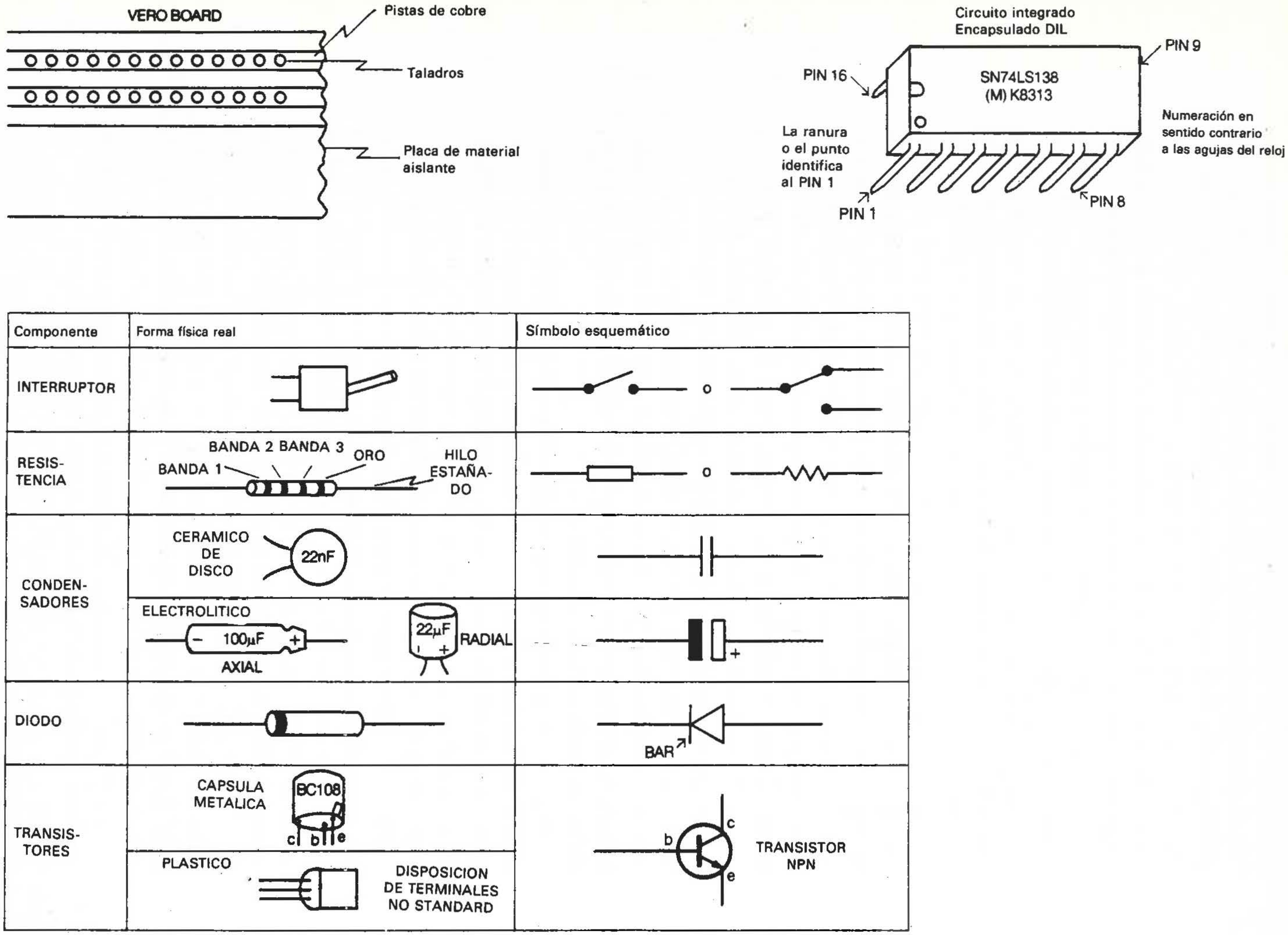


Fig. A.1. Componentes.

Interruptores

El interruptor ilustrado es un interruptor convencional de dos estados estables. Cuando se desplaza el mando a un lado o a otro, se establece el contacto o se deshace, respectivamente entre uno de sus terminales y otro. El diagrama muestra dos tipos comunes de contacto. El segundo de ellos se denomina conmutador de dos posiciones, un circuito, y tiene tres terminales. Los interruptores pueden tener varios conjuntos de contactos accionados por el mismo mando. También pueden tener un mando de más de dos posiciones.

Resistencias

Las resistencias se utilizan para muchas aplicaciones en electrónica. En general, las resistencias más utilizadas son las de cuarto de vatio, de carbón con tolerancia del valor real y el nominal del 5%. Puede haber otro tipo de resistencias, como las bobinadas, pero, al tener una cierta inductancia, pueden interferir el normal funcionamiento del ordenador.

La resistencia que aparece en la figura está pintada con el código de colores más normalmente empleado para su identificación. Para el tipo de resistencia antes mencionada, se utiliza una de las bandas del extremo de color dorado, que indica cuál de los dos extremos de la misma es por el que debe empezar a leer el código, y también indica que la tolerancia es del 5%. Las otras bandas de colores dan el valor de la resistencia de acuerdo con el código que se muestra en la siguiente tabla:

Color de banda 1 ó 2	*Valor*
Negro	Ø
Marrón	1
Rojo	2
Naranja	3
Amarillo	4
Verde	5
Azul	6
Violeta	7
Gris	8
Blanco	9

Estos valores indican el número de ceros detrás del valor indicado por las bandas 1 y 2, cuando están en la banda tres, excepto si ese color es el negro, que indica "no color" en dicha banda tres.

Los valores de las resistencias se expresan en ohmios. Los ohmios normalmente se indican también con una letra "omega" mayúscula. Para simplificar la clasificación, los miles de ohmios se expresan con una K, y los millones con una M. Las fracciones de K se escriben a menudo intercalándola en el lugar de la coma decimal. Así, una resistencia de 2,7 K se suele expresar como 2K7. Lo mismo ocurre con las fracciones de M.

Ejemplos del uso del código de colores son:

Banda 1	*Banda 2*	*Banda 3*	*Valor*	
Marrón	Rojo	Negro	1 2 —	= 12 ohms
Marrón	Rojo	Marrón	1 2 ø	= 12ø ohms
Amarillo	Violeta	Rojo	4 7 øø	= 4K7
Marrón	Negro	Amarillo	1 ø øøøø	= 1øøK
Marrón	Negro	Naranja	1 ø øøø	= 1øK
Amarillo	Violeta	Verde	4 7 øøøøø	= 4M7

Se puede obtener cierta experiencia en el uso del código de colores midiendo resistencias con el multímetro.

Condensadores

Los condensadores se usan para almacenar energía electrónica. Las tensiones de alimentación pueden no ser suficientemente constantes, o continuas, y de hecho siempre tienen un rizado mayor o menor. El uso principal que se ha hecho en este libro de los condensadores es filtrar este rizado.

Los condensadores están marcados en su superficie. La unidad de medida es el *faradio*. Sin embargo, como esta unidad es muy grande en las utilizaciones normales, habitualmente los condensadores que se emplean más frecuentemente se denominan por su valor en microfaradios (μF), nanofaradios (nF) o picofaradios (pF), que son, respectivamente, 10^{-6}, 10^{-9} y 10^{-12} faradios. El voltaje máximo que pueden soportar es especialmente crítico en el caso de los condensadores electrolíticos. También en éstos es importante el sentido o polaridad de su montaje.

Si tratamos de medir la resistencia de uno de estos dispositivos

veremos que es infinita. Por un condensador la corriente continua no pasa, sólo pasa la alterna.

Diodos

Los diodos tienen la característica de conducir la corriente eléctrica solamente en una dirección. Esta característica se utiliza mucho para evitar que cierto tipo de datos lleguen a un determinado circuito, y sí lleguen a otros. También se utilizan para rectificar la corriente alterna. El diodo ha sido el primer tipo de semiconductor descubierto y es el precursor de los transistores y otros dispositivos semiconductores modernos. El símbolo empleado en los diagramas para representar el diodo es una flecha con una barra, que es una identificación acertada del sentido en que puede circular la corriente a través de él. La Figura A.2 muestra de forma esquemática el funcionamiento de un diodo. Cuando se conecta la batería con el elec-

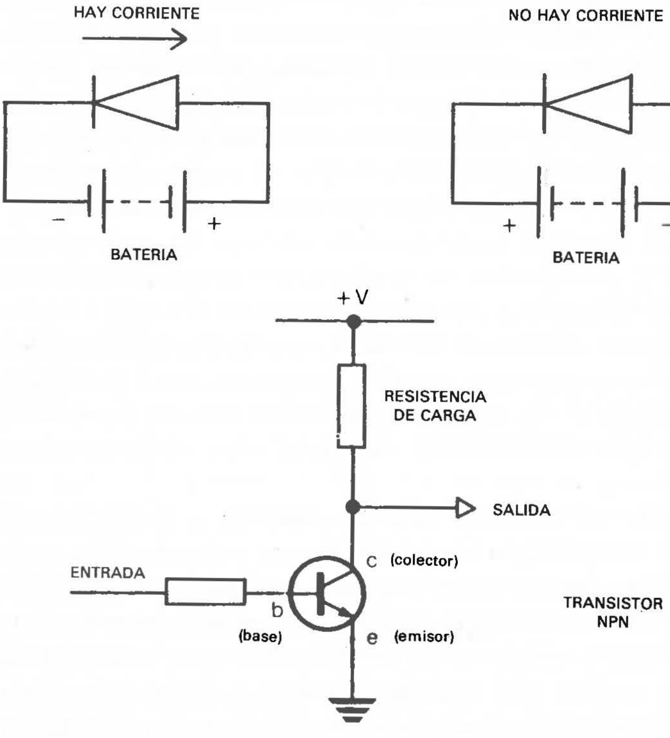


Fig. A.2. Díodos y transistores.

trodo negativo en la barra del diodo, éste se hace conductor y permite el paso de la corriente desde un electrodo al otro de la batería. Cuando la polaridad de la batería se invierte, la corriente no pasa.

Se puede utilizar el multímetro para verificar el funcionamiento de un diodo. Colocado el multímetro en posición de medida de resistencias se observará que midiendo la del diodo, éste tendrá mayor resistencia en un cierto sentido que en el otro cuando el aparato se conecte. Esto no puede apreciarse en ciertos tipos de diodos que tienen una característica tal que las pequeñas tensiones aplicadas en sus extremos no son suficientes para hacerlos conducir. Para verificar uno de estos diodos (por ejemplo, ciertos diodos LEDs) se necesita disponer de multímetros con una batería interna de más alta tensión.

Transistores

Hay muchos tipos distintos de transistores, y dos estilos de encapsulado se muestran en la figura A.1. Un transistor tiene tres tipos de terminales, el *colector,* la *base* y el *emisor,* como se muestra en la figura, con las letras de identificación c, b y e. El tipo de transistor ilustrado es NPN, que significa que el colector ha de ser positivo con respecto al emisor para que el dispositivo funcione. En electrónica digital los transistores se utilizan simplemente como interruptores, cuando se aplica señal a la base. El circuito mostrado en la Figura A.2 incluye dos resistencias y es el circuito común utilizado en este libro. El emisor se conecta a tierra, o Ø V, y cuando la entrada es también Ø V, el transistor no conduce, con lo que la salida está a potencial de la batería, a través de la resistencia de carga. Cuando la entrada se pone positiva, el transistor conduce y conecta la salida a tierra. La salida, por tanto, tiene un estado opuesto a la entrada en todo momento, por lo que a este circuito se le llama inversor. Este circuito es muy utilizado para comandar relés, como se vió en el Capítulo 2.

La teoría del transistor es muy compleja y se sale fuera del objetivo de este libro, pero la noción dada es suficiente para ayudarnos en la comprensión de los montajes a transistores utilizados en este libro. Los transistores se utilizan para muchísimas aplicaciones, pero su uso principal es como amplificador. Sólo una señal muy pequeña se necesita para hacer conductor a un transistor, y el cambio de la tensión en la salida puede ser muchísimo mayor. La resistencia de entrada se necesita frecuentemente para limitar la corriente que puede extraerse de la señal de entrada.

Circuitos integrados

La Figura A.1 muestra un típico circuito integrado (CI) o chip. Este tipo de circuito integrado se denomina un DIL (Dual In Line), por la forma de disposición de sus terminales (en doble fila). El diagrama muestra la orientación del chip y la numeración ordinaria de sus terminales. Esta orientación es la misma para todos los circuitos integrados que aparecen en este libro, sean éstos de 16 o de 40 terminales. En general, es conveniente utilizar zócalos para conectar un DIL en los circuitos, con lo que se evita soldar sus terminales directamente al circuito impreso. Con el uso de zócalos se facilita extraordinariamente el cambio de CIs para reutilización en otros montajes, o en caso de avería.

Siempre hay que ser cuidadoso cuando se inserten CIs en sus zócalos, para no doblar sus terminales o dejar sin conexión a alguno de ellos. Esta es una causa frecuente de anomalías en el funcionamiento de circuitos que utilizan este tipo de componentes.

Para sacar un CI de su zócalo se debe utilizar un pequeño destornillador introduciéndolo entre el cuerpo de éste y su zócalo. Haciendo una pequeña presión o palanca deberá separarse de él hasta que los terminales salgan casi totalmente de sus alojamientos.

Siempre es buena práctica verificar las tensiones de alimentación en los zócalos antes de insertar los CIs en los mismos. Se hará con el multímetro colocado en disposición de medida de tensión. Se deberá verificar que las tensiones en los zócalos son las correctas antes de insertarlos.

Placa de pistas impresas

Uno de los métodos más empleados en la construcción de prototipos de circuitos es la utilización de una placa aislante con pistas de cobre impresas paralelas. El más frecuentemente empleado es el material llamado Veroboard. Se trata de una placa aislante con taladros situados en forma matricial con una separación entre ellos de Ø,1 pulgadas, y pistas de cobre adheridas al lado inferior de la placa. Los taladros atraviesan las pistas de cobre, de manera que se pueden insertar los terminales de los componentes a través de ellos, y soldarlos por el otro lado al cobre de las pistas. Este es un método excelente de fijar los componentes a la placa y efectuar su conexión por medio de soldadura. La placa Veroboard está ilustrada en la Figura A.1.

El uso de Veroboard está descrito en los proyectos de este libro. Los siguientes apartados proporcionan algunos consejos prácticos a aquéllos que no estén familiarizados con la soldadura y los métodos de montaje de circuitos electrónicos.

LEDs y relés

El LED (diodo emisor de luz) tiene las mismas propiedades que los diodos mencionados anteriormente, excepto que emiten luz cuando por él pasa una corriente. La Figura A.3 muestra el símbolo y la forma real del LED. Puede utilizarse una pequeña batería para verificar su funcionamiento en función de la polaridad de la misma, utilizando el circuito de la figura. No es peligroso aplicar tensión con la polaridad equivocada, pero se debe colocar una resistencia en serie con la batería cuando se aplique la tensión de forma correcta para limitar la corriente que puede circular, que si es muy alta podría

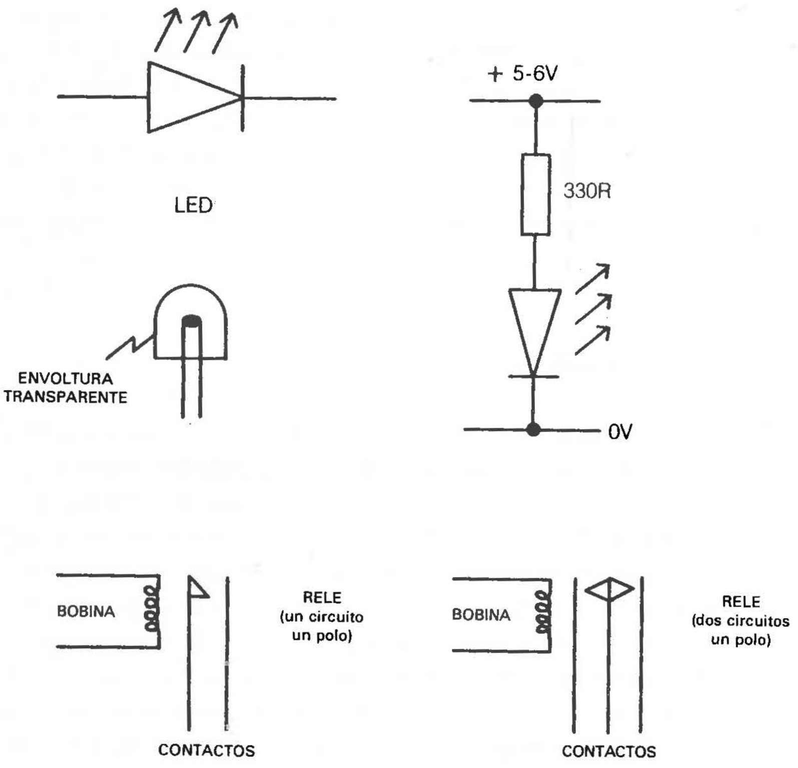


Fig. A.3. LEDs y relés.

destruirlo. La tensión no debe ser mucho mayor que la indicada en la figura, a menos que se aumente la resistencia en serie. Usaremos el multímetro en la escala de corriente para verificar que la corriente que pasa a través del LED no supera el valor de 15 mA aproximadamente.

El LED está alojado en una envoltura transparente, como se muestra, y se suministra en varios colores posibles.

Los relés se explican en el Capítulo 2, y la Figura A.3 muestra dos símbolos utilizados en los esquemas para identificar este componente. Una versión aparece como una bobina de hilo conductor con una pareja de contactos. La otra versión muestra la misma disposición pero con los contactos dispuestos como conmutador. Un lado estará normalmente cerrado, mientras que el otro estará abierto. Cuando la corriente circule por la bobina, la situación será justamente la inversa.

Montaje y soldadura

Para practicar en el montaje de dispositivos electrónicos necesitaremos una placa Veroboard, unos pocos componentes tales como resistencias, condensadores y zócalos de circuitos integrados. También se deberá disponer de las tres clases de hilo de cableado que se han mencionado ya.

Primero habrá que practicar con los alicates de cortes y pelahilos. Se deberá tratar de cortar hilos de distintas clases y descubrir los extremos de su aislante sin cortar el hilo propiamente dicho. Esto es particularmente difícil cuando se manejan hilos para wrapinado. No es fácil quitar el aislante sin romper el hilo. Existen unos pelahilos especiales que deben comprarse para facilitar esta tarea. Con un poco de práctica y la herramienta adecuada esto no es un problema. Se debe adquirir un pelahilos ajustable al grosor del cable. Con dicha herramienta se consigue cortar el aislante sin llegar a cortar el conductor con toda facilidad. Además, habrá que adquirir una herramienta para efectuar el enrollado del cable para el wrapinado. Esta herramienta es barata y utilísima.

Los hilos de gran diámetro no son dificultosos. Para los delgados la cosa puede ser más complicada.

El paso próximo es el aprendizaje de la soldadura. Practicaremos soldando componentes en la placa Veroboard. Primero, no obstante, aprenderemos a utilizar el soldador.

Una vez encendido el soldador y después que éste haya adquirido

la suficiente temperatura, se debe aplicar un poco de estaño a la punta. Si el soldador está suficientemente caliente, la soldadura debe fluir y recubrir la punta del soldador. Esto la protegerá de oxidaciones. A continuación, trataremos de soldar sobre las pistas de cobre de la placa Veroboard. Se debe aplicar la soldadura —el estaño— entre el cobre y la punta del soldador; éste debe presionarse suavemente contra la pista de cobre. La soldadura se habrá hecho correctamente si el estaño fluye sobre la pista de una manera rápida. Será difícil que el estaño empiece a fluir si la pista de cobre está oxidada o tiene algo de grasa por haberla tocado con las manos. Antes de efectuar ninguna soldadura se debe limpiar con algún producto desengrasante. No se debe mantener la punta del soldador mucho tiempo sobre la pista, pues puede estropearse. Esta es una regla general para soldar todo tipo de componentes. Se debe insistir en realizar la soldadura en el mínimo tiempo posible y, por tanto, seguir practicando sobre la placa Veroboard hasta conseguir la suficiente práctica. La soldadura debe quedar limpia, pulida y sin burbujas. De vez en cuando hay que limpiar la punta del soldador. El mejor método es pasarla por la superficie de una esponja húmeda. Cuando la punta del soldador tiene un exceso de estaño se debe sacudir con un golpe seco en el aire, de manera que dicho exceso caiga sobre un papel para evitar realizar soldaduras con burbujas.

Lo siguiente es aprender a estañar los terminales de los componentes y los extremos de hilos de conexión. Estañe siempre la punta del soldador antes de efectuar cualquier soldadura. Compruebe que el terminal o el hilo esté limpio y efectúe la soldadura disponiendo sobre el hilo el estaño y el soldador al mismo tiempo en la forma que muestra la Figura A.4. No es bueno dejar una gran gota de estaño en el terminal o hilo, antes bien, éste debe quedar cubierto en toda su superficie con una finísima capa de estaño. Practique esto hasta conseguirlo en el menor tiempo posible. Un calentamiento prolongado del terminal de un componente puede dañarlo.

Cuando se trata de estañar un cable de conexión del tipo trenzado, o cablecillo, después de pelar el extremo que se quiere soldar, se deben retorcer las hebras que componen el conductor sobre sí mismas y proceder a su estañado como en el caso de cable normal de un solo conductor rígido. El estaño deberá impregnar uniformemente todas las fibras del hilo.

El proceso de estañado debe ser siempre la primera cosa a realizar antes de efectuar cualquier soldadura. Si el soldador ha estado inactivo durante un tiempo aproximado de unos 20 segundos, también debe ser estañado. Una indicación de la necesidad de estañar la

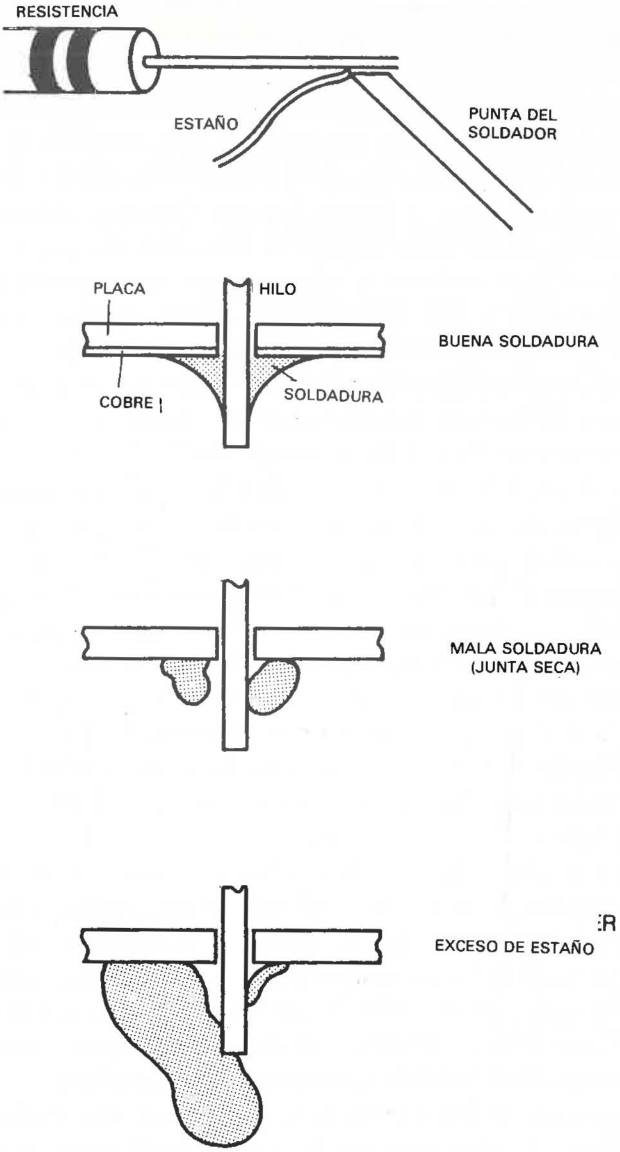


Fig. A.4. Soldadura.

punta del soldador es que deja de producir humo cuando el fundente que incorpora en su interior el estaño se ha evaporado totalmente.

El hilo ya estañado deberá ser pasado a través de uno de los taladros de la placa Veroboard, colocando ésta de manera que veamos las pistas de cobre, y el extremo del hilo que la atraviesa. La placa Veroboard no necesita ser estañada antes de hacer la unión.

Para realizar la soldadura de un extremo de cable o terminal de algún componente, una vez pasado por el agujero correspondiente, se estaña la punta del soldador de la manera descrita anteriormente y se coloca de manera que haga contacto físico con la pista de cobre y con el hilo al mismo tiempo. Entonces se aplica un poco de estaño a la unión, con lo que éste se repartirá entre el terminal y la pista, de la manera que aparece en la Figura A.4 como "buena soldadura". Si el estaño no fluye correctamente o no se han calentado previamente las superficies a unir, ocurrirá el segundo caso mostrado en la figura, "soldadura seca" o fría. Si, en cambio, se ha utilizado demasiada cantidad de estaño, puede ocurrir el caso número tres.

Un tipo de dificultad para una unión que va a ser soldada es que el diámetro del hilo sea mucho menor que el agujero. Este puede ser el caso del hilo utilizado para wrapinar. Se debe ensayar este tipo de soldadura hasta obtener buenos resultados. Conviene asegurarse de que el hilo se mantiene en su lugar durante la soldadura, sin moverse y sin que pase a través del taladro parte de su camisa aislante.

El siguiente componente que se debe practicar es con los zócalos de los circuitos integrados. Debe, en primer lugar, verificarse que la separación entre taladros de la placa se corresponde con la de los terminales del zócalo. Luego se insertará en su lugar el zócalo pasando todos sus terminales a través de los taladros y curvando dos de los correspondientes a las esquinas para que se mantenga en su posición sin caerse. Se supone que ya se han cortado las pistas que unen un lado y el otro del zócalo.

Una vez insertado el zócalo se dará la vuelta a la placa y se aplicará estaño de la manera indicada anteriormente a dos de los terminales situados en esquinas opuestas. A continuación, se mantendrá apretado el zócalo contra la placa, tocando una de las soldaduras realizadas con el soldador colocado de manera que el zócalo quede lo más pegado posible a la placa. Se hará lo mismo con la otra, y luego con el resto de los terminales, con cuidado de no aplicar estaño a dos terminales a la vez para no producir cortocircuitos, cosa no tan fácil, por estar muy juntos. Cada soldadura debe verificarse con una lupa. Si se observa estaño intermedio entre dos terminales, se debe aplicar de nuevo el soldador a uno de ellos para que el exceso vaya a dicho terminal. La inspección de las soldaduras será más fácil si se limpian los restos de fundente que siempre quedan en la superficie de alrededor de cada unión.

Con la práctica, conseguiremos realizar buenas soldaduras sin cortocircuitos, sin exceso de estaño y sin utilizar un calentamiento grande que pueda dañar los componentes.

Apéndice 3
El Conector posterior del Spectrum

La Figura A.5 muestra una gráfica de disposición de los terminales del conector posterior del Spectrum, de una manera un poco más clara que la que aparece en el manual del Spectrum, que es una visión desde un extraño ángulo. Deberemos adquirir el conector correcto y emplear siempre la correcta posición del mismo, ayudándonos siempre con la ranura de que dispone.

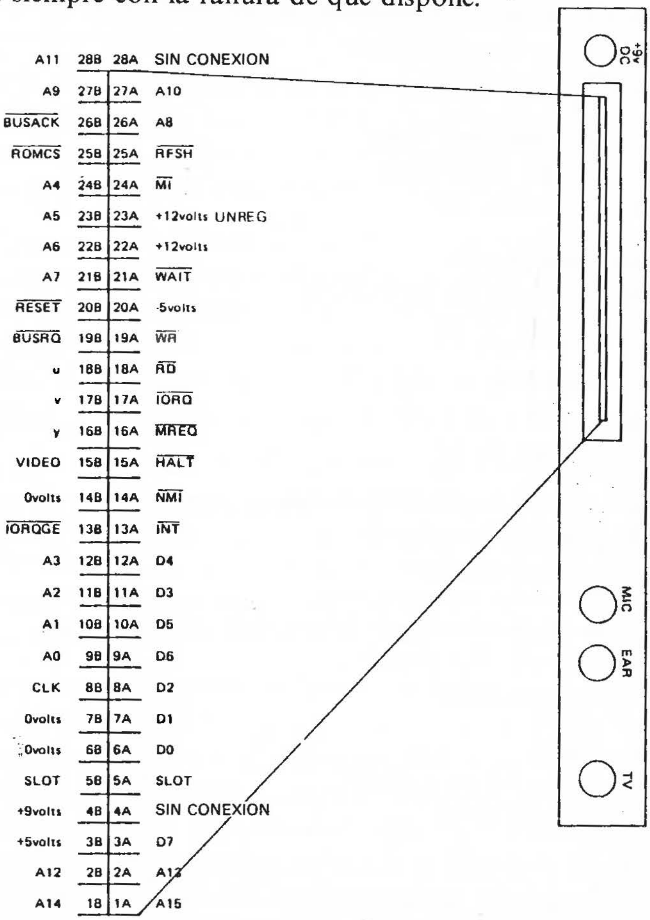


Fig. A.5. Conector posterior del Spectrum.

Apéndice 4
Suministradores de Componentes

Electroson, S.A.
C/ Duque de Sesto, 15. MADRID
Tfno.: (91) 431 14 80

Compañía Española de Suministros Electrónicos, S.A.
C/ Barquillo, 25. MADRID
Tfno.: (91) 232 36 44

Componentes Santos del Valle.
C/ Galileo, 56. MADRID
Tfno.: (91) 446 82 00

Electrónica Buen Suceso.
C/ Buen Suceso, 20. MADRID
Tfno.: (91) 248 03 29

Eectrónica Sandoval.
C/ Sandoval, 4. MADRID
Tfno.: (91) 445 18 33

Actividades y Componentes Electrónicos, S.A.
C/ Maudes, 15. MADRID
Tfno.: (91) 254 68 03

Compañía de Productos Electrónicos, S.A.
C/ Sagasta, 18. MADRID
Tfno.: (91) 446 40 00

Electrónica Lugo.
C/ Barquillo, 40. MADRID
Tfno.: (91) 419 87 51

Sonytel, S.A.
C/ Clara del Rey, 24. MADRID
Tfno.: (91) 416 01 47

Glosario

AC (Alternative Current): Corriente alterna. Con este término se describe una corriente eléctrica, o una tensión que oscila continuamente y cambia de polaridad, como concepto opuesto al de corriente continua (DC). Una tensión que oscila, pero que no cambia su polaridad, se dice que tiene "rizado", más que decir que es corriente alterna.

Activo bajo: Una línea de señal es activa baja si su estado activo es aquél que ocurre con el nivel Ø. Un ejemplo puede ser la línea Enable de un interface o de un bloque de memoria. Para activar tal dispositivo, su línea Enable deberá estar al nivel de tierra.

ADC: Convertidor analógico digital. Un dispositivo que convierte continuamente cambios de tensión a números binarios para la entrada de un ordenador. Una tensión que varía continuamente, o analógicamente, no tiene valores preestablecidos o discretos, pero puede tomar valores comprendidos entre varios niveles dados, o dentro de un rango.

Analógica: Una señal que es capaz de tomar valores variables continuamente se dice que es una variable analógica. Por ejemplo, un sensor de temperatura proporcionará una tensión proporcional a la temperatura. Los valores de tensión para diferentes temperaturas serán capaces de tomar cualquier valor. Este concepto es el opuesto a *digital.*

ASCII (American Standard Code for Information Interchange): Es un código numérico para un conjunto de letras, números y símbolos generales. Un código ASCII es normalmente un número comprendido entre Ø y 127 y, por consiguiente, puede ser representado por un número binario de 7 bits. Sin embargo, puede ser completado ocupando la totalidad de los ocho bits de una palabra, llegando a representarse un total de 256 caracteres diferentes.

Binario: Expresado de forma que sólo utiliza dos niveles, 1 y ∅. Un número binario está formado únicamente por unos y ceros.

Bit: Contracción de las palabras inglesas "binary digits". El uno y el cero son los dígitos binarios. Los unos y ceros de un número binario se denominan *bits.*

Bus: Colección de líneas electrónicas agrupadas que realizan una misión común.

Bus de control: Colección de líneas electrónicas que realizan funciones generales de control para conseguir que el CPU controle el sistema electrónicamente. Las líneas RD, WR, IORQ y MREQ son ejemplo de líneas del bus de control. Este bus es un poco difícil de definir en su totalidad, ya que cada CPU tiene sus líneas de control dispuestas de distinta manera, y algunas líneas de control ni siquiera forman parte de este bus.

Bus de datos: Colección de líneas que transportan los bytes de datos desde o hacia la CPU. Cuando el contenido de una localización de memoria o I/O es solicitado por el CPU.

Bus de direcciones: Grupo de líneas electrónicas que establecen la conexión con las direcciones de memoria en virtud de un conjunto de palabras binarias. Las direcciones son salidas del CPU y están establecidas en conexión con los dispositivos que han de dialogar con él. Unos circuitos lógicos llamados decodificadores de direcciones son los responsables de reconocer estas direcciones y activar los bloques de circuito adecuados. Este primero utiliza el bus de direcciones para contactar con el dispositivo correcto y localizarlo. Los datos de esta localización se colocan en el bus de datos y se recogen por la CPU. El proceso es el mismo cuando la CPU necesita escribir datos en una localización.

Byte: 8 bits forman un byte, y un microprocesador de 8 bits, tal como el Z8∅, dispone sus localizaciones de memoria para contener bytes. Un byte puede contener una palabra completa del código ASCII y, por consiguiente, puede contener un carácter. Sin embargo, para la aritmética de coma flotante, hay que utilizar más de un byte para representar un número general, para obtener la suficiente precisión.

Carta de flujo: Bloques gráficos que indican el sentido lógico de actuación de un programa sin necesidad de comprender el lenguaje del ordenador.

Chip: Circuito integrado. Con la palabra chip se suele denominar el componente físico. Muchos transistores, resistencias y condensadores pueden formar parte de un chip, unidos por medio de diminutas conexiones. Los componentes y sus conexiones se realizan por medios fotográficos y químicos.

Circuito integrado: Encapsulado de pequeñas dimensiones, dentro del cual hay un circuito electrónico completo formando un chip de silicio. La integración es el proceso de componer el chip que contiene el circuito, en vez de utilizar los componentes discretos, tales como transistores, resistencias, etc.

Código máquina: El lenguaje real de la CPU. Las instrucciones se almacenan como números binarios que son interpretados por la CPU directamente. La CPU nunca ejecuta directamente las órdenes BASIC. El ordenador siempre tiene que tomar las órdenes de control del teclado, o sacar resultados por la pantalla traduciendo las instrucciones BASIC al código máquina correspondiente. El Spectrum tiene un programa llamado un "intérprete", para ello.

Condensador: Componente electrónico que almacena carga y que no permite el paso de la corriente continua. Los condensadores se utilizan frecuentemente para filtrar la salida de las fuentes de alimentación. En este libro se han utilizado dos tipos de condensadores: el electrolítico y el disco cerámico. El condensador electrolítico es capaz de almacenar más carga que el de disco y debe conectarse con su polaridad correcta. Los condensadores cerámicos son de pequeño valor y se utilizan para eliminar el ruido de alta frecuencia en las fuentes de alimentación. Ambos tipos de condensadores se utilizan también para bloquear el paso de tensiones continuas.

Controlador de motores paso a paso: Dispositivo electrónico que convierte la salida de un ordenador en señales de control para un motor paso a paso.

CPU: Unidad de proceso central. La CPU de un ordenador es el centro aritmético y lógico de la máquina. Además, es responsable del control de toda la electrónica del sistema. En los grandes ordenadores, la CPU frecuentemente ocupa una serie de tarjetas de circuito impreso. En un microordenador, toda ella está confinada en un solo chip, la MPU, es decir, el microprocesador. Por ello se usan ambas denominaciones, CPU y MPU en el argot de los microordenadores.

DC (Direct Current): Corriente continua. Con este término se designa la salida de una fuente de tensión cuyo voltaje no cambia de polaridad, al contrario que la corriente alterna. Una batería es una fuente común de DC.

Decimal: Nuestro sistema de numeración normal basado en el número 1Ø. Los números de este sistema se llaman números decimales y al sistema se le denomina a veces sistema denario.

Digital: Se refiere al empleo de niveles discretos de valores que puede tomar una variable, y es lo opuesto a *analógico*. Cualquier variación que tiene un conjunto de niveles discretos posibles se llama digital. Por ejemplo, un motor paso a paso es un dispositivo digital, porque su movimiento es a saltos precisos, no continuos. No puede tomar una posición intermedia entre la de dos saltos consecutivos.

Diodo: Componente electrónico que bloquea las corrientes que fluyen en una dirección, dejando pasar las de la otra dirección. Hay muchos tipos de diodos y muchas aplicaciones diferentes para ellos. En este libro se utilizan para rectificar la corriente alterna y como protección de los transistores que gobiernan relés. También se emplean unos tipos especiales de diodos emisores de luz (LEDs), como lamparitas.

Enable: Línea electrónica que se utiliza para activar un dispositivo. Los dispositivos de memoria tienen líneas Enable, para activarlas individualmente y permitir que a todos los bloques llegue la misma información, lo que ahorra líneas de datos en el diseño de ordenadores. Normalmente son líneas activas bajas, es decir, se activan cuando el nivel es bajo en ellas.

I/O: Entrada/Salida. Dispositivos electrónicos de un ordenador que permiten al sistema comunicarse electrónicamente con el mundo exterior. En ellos están incluidos, por ejemplo, los conectores MIC y EAR y el zumbador del Spectrum.

LAS (Light-Activated Switch): Interruptor activado por luz. Este dispositivo abre y cierra un interruptor cuando la luz que incide sobre él alcanza un determinado nivel.

LDR (Light-Dependent Resistor): Resistencia dependiente de la luz. Este dispositivo cambia su resistencia de acuerdo con el nivel de luz incidente. Una forma corriente de LDR es la célula de sulfuro de cadmio. La variación de la resistencia es una función inversa a la variación de la luz. A más luz, menor resistencia.

LSI (Large Scale Integration): Integración a gran escala. Se refiere a los dispositivos que contienen grandes densidades de componentes integrados dentro del chip. El microprocesador y el PIO son ejemplos de dispositivos LSI.

Mapa de memoria: Conjunto de direcciones de memoria. Modo como están dispuestas estas localizaciones, numeradas para que el CPU pueda establecer contacto con las distintas localizaciones.

Mapa I/O: Conjunto de direcciones de los dispositivos I/O, o la manera en que están dispuestos, numerados, al igual que los bloques de memoria.

Memoria: Dispositivos electrónicos que guardan instrucciones y datos. Hay dos tipos de memoria mencionados en este libro, cada una de las cuales se contienen en diferentes chips: la memoria RAM y la ROM. Se denominan, a veces, memorias de semiconductor parA distinguirlas de las memorias de cinta magnética o de disco.

Microcomputador: Microordenador. Es un sistema que se basa, fundamentalmente, en el empleo de un microprocesador y memorias asociadas. Se pueden obtener microordenadores con sus memorias y dispositivos I/O en un solo chip.

Microinterruptor: Interruptor miniatura muy sensible que requiere muy poca energía para activarlo.

Motor: Máquina eléctrica que se basa en los fenómenos electromagnéticos para girar cuando se le aplica energía eléctrica.

Motor paso a paso: Motor cuyo eje gira en saltos discretos que forman un determinado ángulo. No giran continuamente como un motor convencional, por lo que requieren un dispositivo que los controle, consiguiéndose un movimiento muy preciso y controlable.

Multímetro: Aparato de medida para voltajes, corrientes y resistencias.

MPU: Microprocesador. Circuito integrado, corazón del microordenador. Es capaz de buscar a través de la memoria, ejecutar instrucciones y controlar el sistema a través de los buses.

Paralelo: Comunicación a través de varias líneas electrónicas simultáneamente. Las líneas de salida del PIO son líneas paralelo.

PCB (Printed Circuit Board): Placa de circuito impreso. Se utiliza para fijar e interconectar los componentes electrónicos. La interconexión se realiza por medio de pistas de cobre impresas sobre una placa de material aislante, que suele ser fibra de vidrio. Los componentes se fijan metiendo sus terminales a través de taladros, soldándose a las pistas por medio de estaño de soldadura.

PIO: Chip paralelo I/O. Es un circuito integrado de la familia del Z8Ø que dispone de 16 líneas I/O en paralelo para añadir al ordenador. El microprocesador se comunica con él por medio de algunos registros internos que tiene el PIO.

Placa de pistas impresas: Un tipo de circuito integrado de uso general que dispone de taladros separados de manera standard y con pistas de cobre paralelas sobre la placa de fibra de vidrio.

Potenciómetro: Resistencia variable.

PSU (Power Supply Unit): Fuente de alimentación.

Radiador: Lámina de metal, frecuentemente aluminio, a la que se fijan rígidamente aquellos componentes que necesitan disipar calor producido por el paso de la corriente a su través.

RAM (Random Access Memory): Memoria de acceso aleatorio. Este tipo de chip de memoria almacena información que se ha escrito en ella y que puede extraerse de ella. Es una memoria volátil, es decir, la información contenida en ella desaparece cuando no está alimentada por la tensión correspondiente.

Realimentación: Proceso por el cual el dispositivo a controlar informa al controlador de su posición o actuación, después de recibir un comando de control.

Reconocimiento de voz: Es el proceso, todavía embrionario, realizado por ordenador, de reconocimiento de los lenguajes humanos. Es lo opuesto a síntesis de voz.

Registro de control: Es una localización interna de memoria de un dispositivo electrónico a través del cual el microprocesador escribe comandos al dispositivo. Por ejemplo, cuando utilizamos el PIO sus funciones son totalmente controladas por los bits de sus registros de control. Con objeto de poner una línea activa, el bit correspondiente en el registro de datos se pone a 1. Es éste un método general para el establecimiento de comunicaciones con dispositivos I/O de gran integración (LSI).

Regulador: El regulador de tensión es el dispositivo electrónico encargado de estabilizar la tensión de salida de una fuente de alimentación.

Relé: Dispositivo electromecánico, que dispone de unos contactos gobernados por la tensión aplicada a la bobina de que dispone.

Reset: Terminal de un microprocesador para forzarlo a comenzar desde un estado conocido, particularmente cuando se enciende.

Resistencia: Componente electrónico que reduce la corriente.

Robot: Dispositivo mecánico que realiza funciones automáticamente, frecuentemente controlado por realimentación, y con alguna capacidad de aprendizaje. Cualquier dispositivo mecánico que se conecta a un ordenador para que sea controlado por éste, puede ser llamado robot.

ROM: Memoria de sólo lectura. Este tipo de memoria no puede borrarse y es utilizada para almacenar programas para el control del ordenador, que permanece en ella aunque se apague el ordenador.

Sensor: Dispositivo electrónico para medida y comunicación del estado de una variable física hacia un ordenador.

Sensor de posición inicial: Interruptor fijado a una parte móvil de una máquina que se cierra cuando la posición de dicha parte móvil alcanza una posición que interese. Se utiliza como realimentación hacia el ordenador para mostrar que la posición indicada ha sido alcanzada.

Servomotor: Un tipo de motor que utiliza un sensor posicional o de rotación para enviar realimentación al sistema de control para monitorizar y, por consiguiente, controlar el movimiento del motor.

Síntesis de voz: Es la síntesis electrónica de la palabra hablada. Las palabras que deben pronunciarse y comunicarse por algunos sistemas en forma de cadenas de caracteres ASCII. Es más común, sin embargo, para un dispositivo de síntesis de voz, tener almacenadas un número de palabras en una ROM, las cuales son llamadas de manera aleatoria para cualquier pronunciación dada.

Solenoide: Bobina de hilo conductor enrollada alrededor de un núcleo de hierro que se utiliza para crear un campo magnético cuando pasa una corriente eléctrica a través de ella.

SPC (Speech Processing Chip): Chip procesador de voz. Se utiliza para recuperar en un momento dado los datos necesarios para reproducir la voz humana. Dichos datos han sido previamente grabados.

Transformador: Dispositivo eléctrico que se utiliza para elevar o disminuir el voltaje eléctrico sin pérdida apreciable de potencia.

ULA (Uncommited Logic Array): Conjunto lógico diseñado por encargo. Es un chip que debe ser definido al constructor para contener un circuito electrónico dado. Cuando una empresa tiene un gran número de chips conectados juntos para formar un circuito, éste puede ser encargado para ser realizado totalmente en un solo chip por un fabricante adecuado. La ULA en el Spectrum es un circuito de éstos, que realiza muchas funciones para el microprocesador.

Vídeo: Se aplica esta expresión a la parte del ordenador que se ocupa de la representación de datos en la pantalla.

Z8ØA: Código industrial del microprocesador particular que se usa en el Spectrum.

Indice